

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/01 11:38:24	This is a test talk title	Max 1,500 characters. A talk is more likely to be accepted if it is Android specific, has expertise and innovative content.	その他 (Other)	Form test	30分 / 30 minutes	DroidKaigi staff
2016/10/01 12:22:29	全てSIになる～RxJavaとLWSを持ち込む楽しさ～	<p>■ 対象者 初級～中級者向け</p> <p>■ 内容 Android 7.0 SDKからStreamやOptionalなど、Java 8で導入されたAPIが一部使えるようになりました。ただし、これは応募時点(2016/10/01)ではJackコンパイラを使う場合に限られていて、実践導入はまだ先になりそうです。そこで、以前から話題となっているRxJavaや、Stream/OptionalのバックポートライブラリであるLightweight Stream APIを活用して、来たJack時代のAndroidアプリ開発について先取りできるお話したいと思います。</p> <p>プリミティブなAndroid SDKのAPIを使う場合と比較してどのようなメリットがあるのか、デメリットは何かあるのかといった話や、RxJava/Lightweight Stream APIを実践導入して得られた知見を皆様と共有できればと思います。</p> <p>特にLightweight Stream APIに関しては単なるバックポートではなく、独自の機能も持っていて、Android開発において特にFragmentを使うときにちょっと便利なユーティリティとして機能します。Android開発においてはメリットがあるライブラリだと思いますので、この知見については一番共有できればと思っています。</p> <p>対象者: イベントのスタッフや運営に興味のある方</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	ryugoo
2016/10/01 17:40:51	カンファレンス向けガイドアプリの作り方	<p>9/20-24に行われた PyCon JP 2016 では、初めてiOS向け/Android向けの公式ガイドアプリを作りました。このうち、Android版公式ガイドアプリの開発経験の中で得た、カンファレンス向けアプリの作り方と注意点を話します。</p> <p>* Android版PyCon JP 2016公式ガイドアプリ * Playストア: http://goo.gl/E9LMhj</p> <p>大まかなセッションの流れとしては、下記のような構成を考えています。</p> <p>* 導入 * アプリを作ることになったきっかけ * アプリの簡単な紹介 * 使ったライブラリやサービス、開発スタイル * 開発過程 * 機能選定 * デザイン調整 * API(Webサイト)削と調整 * Webビューを使う割り切り * スタッフ内でのベータテスト * リリース * iOSと足並みを合わせたリリース日の調整 * そこから感嘆のアップデートまつり(笑) * リリース日や当日の告知方法について * リリース～当日の利用状況 * GoogleAnalyticsの結果公開 * 参加者からの直接のフィードバックいただいた話 * 当日を終えて * アンケート結果公開 * アンケート結果、フィードバックに対して回答 * まとめ * カンファレンス向けアプリを作る上での注意点 * 必須な機能、入れたほうがいい機能、なくてもいい機能</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	rhororo
2016/10/02 23:34:52	Various Android Coding Style	Android開発が、KotlinやDatabindingなど開発手法が多様化する中で自分のチームにあったコーディング規約を早く選択できることは、チームの開発効率をあげます。このセッションでは、それぞれの開発手法ごとに分けた、コーディング規約の紹介と、それぞれの開発手法を変更する際のコーディング規約の選択についてお話します。また、コーディング規約を定めるだけでなく、どのようにしたら規約をチームに浸透させることができるのか、各種ツールの導入や設定なども含めてお話をしたいと思います。	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	保坂 駿
2016/10/03 2:15:02	Google Daydream: Unity, Web, Java or C++?	<p>After we could lay our hands onto the first Google Virtual Reality Daydream™ certified devices at the end of 2016, let us take a look on how to program for Daydream™: When should we use Unity, Java, C++ or even the Web for our projects?</p> <p>I'll quickly introduce the Daydream™ devices, and will give an introduction to the hardware and the expected behaviour of the new Virtual Reality head and hand set.</p> <p>This talk will emphasise giving a hands-on comparison of the current available technologies to program Daydream apps: At first, we'll go through a simple "hello world" style demo app for unity by following me through the steps needed, from installing Unity to simulating the result in the IDE.</p> <p>After discussing the unity advantages and shortcomings, I'll take a look into how to program a similar app in native Android. Highlighting SDK differences between Unity native, Unity GVR Plugin, Android Java SDK and Android Native C++ SDK.</p> <p>In closing, I'll quickly highlight on how to write a photosphere app in native Android and Web.</p> <p>After my talk, the audience will be equipped with a toolset on deciding which technology is to be used for which kind of Virtual Reality App on Android.</p>	その他 (Other)	A mixture between Tooling/Hard	50分 / 50 minutes	MarioBodemann
2016/10/03 11:17:56	ARアプリ作成の最新事情	<p>2016はVR元年と言われていますが、Pokemon Goを筆頭にARの利用も近年盛んになってきています</p> <p>本セッションではARとは何かWeb, アプリなど様々なARアプリ実行環境についての解説ARアプリを実装する際のようなIDE, SDKが適切かなどを解説致します</p> <p>さらにGoogleが提供するAR端末、Project TangoがなぜARにおいて革新的なのかを説明させていただきます</p> <p>ARについて興味がある方向けです</p>	その他 (Other)	AR	30分 / 30 minutes	Tatsuya Shimada
2016/10/06 4:47:00	Toothpick: a fresh approach to Dependency Injection on Android	<p>You find RoboGuice simple but slow ? And you think Dagger 1 & 2 are fast but complex and bloated ? Toothpick is the best of both worlds !</p> <p>Toothpick is a scope tree based, runtime but reflection free implementation of JSR 330. It is pure Java, with a special focus on Android.</p> <p>Toothpick is fast (even faster than Dagger 2 in some cases!) and is simpler to use, with less boilerplate code. Its syntax is very close to Guice. It supports named dependencies, lazy dependencies, providers, and has built-in support for custom scopes. As compared to Dagger 2, Toothpick provides more powerful testing support, allowing you to leverage DI to mock dependencies using either mockito or easymock.</p> <p>Its explicit scope tree helps developers to build more robust apps. It provides more fine grained control to easily implement complex user flows that span multiple activities or state preservation across configuration changes (e.g. rotations), a common issue when implementing the MVP pattern.</p> <p>During this talk for experts, we will introduce Toothpick, its main features and how it compares to other DI libs. We will explain how Toothpick scopes and scope annotations can solve advanced use cases when developing Android apps.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Stephane Nicolas & Daniel Molinero Reguera

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/06 4:50:47	Better Android Intents with Dart & Henson	<p>Intents are an essential component of the Android ecosystem. They are used to express an action to be performed and can be classified into implicit and explicit intents. In an abstract way, all intents together define a navigation layer inside an application.</p> <p>During this talk we will explain why the Android way to create explicit intents is error-prone and also show some problematic ways to solve it. Then, we will introduce the solution we developed at Groupon: Henson, a new library in the Dart project, that takes intent creation to new levels: it generates a fluent API to build intents. This API constitutes a navigation layer that makes it easy, convenient, fast and robust to navigate among your activities and services. Therefore, it will be impossible to miss a required extra and simple to add optional arguments as needed.</p> <p>The talk will be based on the following article featured by Android Weekly: https://medium.com/groupon-eng/better-android-intents-with-dart-henson-1ca91793944b#4h12yxtI</p> <p>Furthermore, all the library features and possibilities will be explained in depth.</p> <p>This intermediate level talk about Dart & Henson will be carried out using slides, code and demos.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Daniel Molinero Reguera & Stephane Nicolas
2016/10/19 11:42:45	Deep dive into Android Auto	<p>■対象 ある程度 Android アプリ開発・リリースの経験のある中級者以上</p> <p>■内容 Lollipop の登場とともに発表された Android Auto は、少しずつ市販車のカーナビシステムに組み込まれてきており、アプリが自動車内にいるユーザーとのエンゲージメントを得るために必須のフレームワークとなることでしょう。そのため、これから Android Auto と連携する機能を構築するには、どのような API があるかだけでなく、どのような仕組みの上に成り立っているかを知っておくことも重要です。</p> <p>本セッションでは、Android Auto そのものの概要やそのフレームワークがもつ API の紹介から、フレームワークがどのように構築されているか、Android Auto を支えている技術を探りて見ていきます。その上で、フレームワークを使うときの注意点やノウハウ、また注意点・ノウハウの Android Auto 以外での一般的な応用を考察します。</p> <p>■構成 * Android Auto の概要 * Android Auto が持つフレームワークの紹介 * Android Auto for Messaging * Android Auto for Audio * Android Auto を支える技術 * Notification * プロセス間通信 * Service * Android Auto フレームワークを使う上での注意点 * プロセス間通信におけるメモリ管理 * Service の死活監視</p> <p>■本セッションに関連する話題 * Android Auto * Notification * Service * プロセス間通信 * AIDL * サポートライブラリ</p>	ハードウェア (Hardware)		50分 / 50 minutes	Keith Yokoma
2016/10/06 12:27:39	Deep dive into video and ExoPlayer	<p>In this talk I'm going to talk about some video technologies and how to deal with them by ExoPlayer, the new standard of Android media player. Unfortunately few developers have a chance to deal with video deeply on their product. But actually video is kind of the most deep side of mobile app development, such as:</p> <ul style="list-style-type: none"> - The difference between streaming formats(DASH, HLS) and container formats(MP4, WebM) - What is DRM, how it works on Android - How to use ExoPlayer and how to customise it - How to support offline playing <p>I'm going to share how I've addressed with them with our product, Quipper Video. Conclusively you'll learn how to manage video and how to use ExoPlayer effectively.</p>	その他 (Other)	Media	30分 / 30 minutes	hotchemi
2016/10/06 13:01:21	Espresso, beyond the basics	<p>More than the typical onView example Espresso offers a complete framework to interact with your UI. Once you start developing your tests you find common issues, in this talk learn tips and tricks on how to deal with the framework and avoid problems like synchronization, permissions, environment test scenarios and much more.</p>	その他 (Other)	Android Testing	50分 / 50 minutes	Inaki
2016/10/06 13:06:35	CIの導入における選択肢と、最高の環境	<p>対象者: CIの導入を考えている方、CIを導入したは良いがポテンシャルを発揮できていないかと思っている方CIとは何なのか、この世にCI as a Serviceと呼ばれるサービスが多数存在すること、テストやビルドが自分で出来る程度の前提知識が必要です</p> <p>内容: これからCIを導入するにあたって、各CI as a Service(ClaaS)などの今考えられる選択肢とその選び方及び、考えうる最高のCI環境とその運用により得た知見について話します</p> <p>最適なCIのあり方というのは開発規模や使っているツール等によって異なります 開発規模や利用ツールによってどのようなClaaS或いはソフトウェアが選択肢にあがるのか？ 各ClaaS或いはソフトウェアの強みや弱み、それらを踏まえたように選択肢していくのが良いのか？ 選んだ環境で、私達はソフトウェアの品質向上のために一体どのような事が可能なのか？ また、現状考えうる最高のCI環境の提起と、いかにしてその選択肢に至ったか？ その運用例と、運用によって得た知見について</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	komatatsu
2016/10/06 16:17:18	Android Wear Essentials	<p>Learning about Android Wear can make you a better Android developer, even if you will never specifically develop for wearables, focusing instead solely on phones and tablets. This is what I realized after developing my latest watch face. During this talk, I will explain the nuts and bolts of developing watch faces and apps for Android wear: from the platform constraints to describing step-by-step how to build an "OMG SO PRETTY!!!" watch face without embedding any images at all. Only via manual draw operations. Ever thought radians, shaders, radial gradients, canvas, paths, xfer modes, shadows, quadratic curves were difficult concepts? These are pretty easy actually, and you'll be able to understand and use those in your Android applications too, after attending this talk</p>	ハードウェア (Hardware)	Android Wear	50分 / 50 minutes	Gautier Mechling
2016/10/06 16:38:04	Develop fast, think scale	<p>It is Android specific, about architecture that easily tested and scale. not a lot of developers use common software engineering pattern. I will present modular development progress.</p>	設計・開発手法 (Architecture and Development Process/Methodology)	Project architecture, MVP, DI, R	50分 / 50 minutes	Leonid Olevsky
2016/10/07 5:12:25	Android MVC architecture: Survival Guide	<p>In the Android environment, there is great flexibility in the organization and architecture of an application. Therefore, it becomes important to use a simple and efficient architecture as MVP. This talk will teach you what is MVP, and why is it important to use as a pattern to create a robust, scalable and easy to test Android application. You will learn the theory behind the concept of MVP seeing the details of its structure to better understand its benefits. The talk will be concluded by a practical example.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Wilfried
2016/10/08 3:51:21	TDD for real world	<p>TDD in Android is not a common practice and is not standard, but following the SOLID principles we can create self contained components that can be testable with the use of JUnit, Mock frameworks, different gradle configurations and UI test suites like Espresso even for testing audio, video, or GPS.</p> <p>MVP and Custom UI Components that are common practices for avoiding God activities can be a good result of doing a good use of TDD.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Cristian Gómez
2016/10/09 5:55:44	Smoke and Mirrors in Android UI	<p>Smoke and mirrors is an expression used to describe something that obscures the truth. The expression is commonly used to describe magicians. Android developers have a lot in common with magicians, they make great and responsive UI, with limited resources. In order to make these experiences possible Android devs need to use a lot of tricks. The talk will show examples of smoke and mirrors in: Android Framework, Google Photos, Google Calendar and Twitter. After the talk, attendees will know how those great experiences are built.</p>	UI・デザイン (UI and Design)		50分 / 50 minutes	Israel Camacho @rallat

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/09 6:25:31	Polyglot Android Development	Historically Android development has been heavily tied to Java. After the Oracle v Google trial, and with the advent of even more viable alternatives, it'll be good to take a look at the state of the ecosystem outside the official tools. The talk will outline the challenges any existing language faces when used in an Android environment. How and why some are able to directly access the SDK, while others require bridging over the NDK. We'll do an overview of many languages, their strengths and weaknesses, and some of the extra tools they bring into the table and are not available in our day-to-day development. Lastly, the talk will cover some multiple possibilities available to share code across platforms without leaving Java by using interpreters, code generators, or transpilers.	その他 (Other)	Alternative Languages	50分 / 50 minutes	Paco Estevez
2016/10/09 6:37:06	Architecture outside Android	Many of our app share functionalities with a web version. But, even if we're solving the same problems, do we know how those web apps are developed? You may have heard before about React, Redux, or the Elm architecture. They have revolutionised the way front-ends are developed, and they come with many guarantees and tools to accelerate development. We'll go over their foundations, pros and cons, amazing tools and libraries, and some of the wisdom we could take from them as a community.	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Paco Estevez
2016/10/09 6:48:00	Pragmatic Functional Android	Functional programming is slowly becoming part of the wider development community. While you may use some tools or libraries that are based on functional principles, it is still uncharted territory for the wider Android community. This talk will cover my personal experiences acquired after years using functional and reactive programming with production apps: what are the real advantages of this new approach? which learning resources apply to Android development? what tools and libraries does it require? how functional does my code have to be to reap benefits? which patterns can you introduce on a legacy app? how do I get started with a reactive app? how do you help your team re-learn development? These and several other questions will be answered.	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Paco Estevez
2016/10/10 0:29:12	Wear 2.0 Complications API	新しい種類のアプリを作りたい人向け Android wear2.0では多くの機能が追加されましたが、Watch face Complicationsなどいくつかの追加機能によりAndroid wear2.0のユーザー体験はそれ以上に大きく変化します。 Watch face Complicationsを使用したWatch faceとData providerの作成方法や、通知を見る、だけに留まらないWearable端末を使った新しいユーザー体験について、考察していきます。	ハードウェア (Hardware)		50分 / 50 minutes	Kenji Matsuoka
2016/10/10 17:58:11	個人で11個のアプリを公開した結果	■概要 僕が個人で開発してきた様々な種類(ツール、ゲーム、Android Wear)のAndroidアプリを公開することによって得られた知見を実際のインストール数などの数値なども交えてお話します。 実際のソースコードというよりは、どんなアプリがインストールされたのか、レビューに対する反応はどうすればよいのか、個人で開発する上で必要なものなどのお話を予定します。 今後の個人開発をする上での参考になれば幸いです。 ■対象者 - 個人でのAndroidアプリ開発に興味がある人 - これから個人でAndroidアプリを開発しようと思っている人 - Androidアプリを開発したいけれど何を開発しようか悩んでいる人 ■目次案 - 個人でAndroidアプリ開発をはじめたきっかけ - 開発してきたアプリについて (インストール数やレビューなど) - ライフスタイル系 - ツール系 - Android Wear系 - ゲーム系 - 個人で開発する上で困りそうなお話 - アプリ公開するためには何が必要? - アプリは開発できるけどアイコンはどうするの? - レビューきたらどうすればいいの? - どんなアプリがインストールされやすいの? - etc...	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	syarihu
2016/10/15 0:03:20	Refactoring Features in Large Codebases	Subject : Advanced Using Android Style to refactor large code bases , to have visual consistency in your application.	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Rahul Dandamudi - Instagram
2016/10/15 11:54:12	Android CI in practice	Based on the AndroidWeekly featured https://medium.com/@Malinskiy/android-ci-with-docker-a2f522086640 article, this talk will walk you through the development processes which will allow you and your team to succeed in a fast-paced Android development with confidence in the quality of the final product	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Anton Malinskiy
2016/10/16 21:26:39	Android ORMの選び方	対象: 初心者~中級者向け Android ORMは古くて新しいテーマです。以前はActiveAndroidやGreenDAO、そしてORMLiteくらいしか選択肢がありませんでしたが、近年は綺麗な星のごとく新しいライブラリが次々に生まれています。また2016年3月に1.0となったRealmというSQLiteベースでないORMも人気を博しています。また発表者はOrmaというORMを開発しており、ORMには一家言ある開発者です。 さて、このORM百花繚乱というこの状況は多様性という意味では嬉しいものですが、逆に選択肢が多すぎて選択するのが難しいという状況となっています。その結果、もっとも有名なORMである、しかし今やメンテナンスされていない古いライブラリであるActiveAndroidが使われ続けるといことも少なくないようです。これは大変残念です。 このセッションでは、Realmを含むさまざまなORMを、インターフェイス・マイグレーション・パフォーマンスについて比較検討します。本セッションがORM選択の一助となれば幸いです。 ## アウトライン (仮) * なぜORMが必要か * ORMは使わない * リリースのキヤンセル * ローカルのデータベース * 素のSQLiteDatabaseの基本 * ライブラリ選り * Simple Helpers * ORMLite * SQLBrite * SQLDelight * O/R Mappers for SQLiteDatabase * ActiveAndroid * GreenDAO * DBFlow * Requery * Orma * Non-SQLite Databases * Realm * インターフェイスの詳細比較 * パフォーマンスの詳細比較 * マイグレーションの詳細比較	その他 (Other)	Library / Client-side Database	50分 / 50 minutes	gfx
2016/10/17 4:52:35	Data binding in the real world	Data binding is not yet widely used by Android developers, and those who do use it seem to limit themselves to just replacing findViewById. However, the possibilities of data binding are endless, and with the right architecture, your code can be much cleaner and a lot easier to understand. This talk begins with explaining the basics of data binding, and then quickly moves on to more advanced techniques/functions. ----- I find that data binding on Android has not gotten enough coverage, and the articles and talks that do appear are very basic and not based on real life usage. I want to make Android developers realize that there is a lot more to data binding than what they have been able to read so far and explain how to architect an app with data binding in mind.	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Kevin Pelgrims

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/17 10:20:39	Data Bindingで開発を気持ちよくしよう	Android 開発はボイラープレートを書くのが辛いんですか？ findViewById、view.setText、view.setOnClickListener、nullチェック等。。。これらを少し楽にしてくれるライブラリはいくつかありますがそれでも完全に解決しません。Beta が取れ、実用段階に入った Android の新ライブラリ Data Binding によって圧倒的に行数を減らしてボイラープレートの問題は解決できるし、Data Binding は一部だけに適用もできるので規模を小さく、安全に少しずつ導入する事が可能です。 下記のネタをカバーします： ・Data Binding 移行の各ステップ ・裏で行われている仕組みの説明 ・Data Binding におけるベストプラクティス このセッションに参加された方は自分のプロジェクトを Data Binding へ移行するのに必要な知識を持ち、Android 開発を気持ちよくできるような人になるはず！	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	ケノド・ブノア
2016/10/18 10:55:31	Optimizing your App UI and View layer	It is important that our Android application could be fully optimized in order to allow a great performance in the whole environment of devices. For this, it is important that not only our application's layout must be well optimized but also that the logic behind the View layer is well coded. There are several when we are implementing the layouts such as "siblings instead of children" and to make this happen, lint tools, hierarchy viewer and the use of particular structures are the best options to achieve this. We'll see some examples and good practices that point to improve the performance of an Android application just optimizing the layouts and tuning the code in the View layer (activities or fragments using SparseArray)	プラットフォーム (Platform)		30分 / 30 minutes	cdmunozi
2016/10/19 11:06:48	未熟なチーム開発	トピック: チーム開発 未熟なチームでどうやってプロジェクトを進めていくのか スタディサプリ English(https://play.google.com/store/apps/details?id=jp.eigosapuri.android)を、自分と新卒とiOSエンジニアの3人で開発していくに当たって、初めにどういふことを考えたか、何がうまくいって、何が失敗したか。 中盤はどうだったか。リリース間際は何をしていたのか。 などなどの、決してスーパースターばかりのチームではないメンバーでどうやってプロジェクトをうまく進めていけばいいのか、試行錯誤した内容をお話します。 agenda(仮) - 前提条件 - 環境 - チーム構成 - 納期 - 開発序盤 - ルール決め - 土台作り - 反省点 - 開発中盤 - 途中からつられたルール - 魔女化 - 新卒を育てる - スキルとタスクの粒度 - 反省点 - 開発終盤 - QA - まとめ	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	kgmyshin
2016/10/20 0:48:02	Xamarin.Android で始めるクロスプラットフォームモバイルアプリ開発	■対象者 ・Android中級者～ ・C# 初心者～ ■概要 Xamarin(ざまりん)は C# によるクロスプラットフォームモバイルアプリ開発ツールです。Microsoft が2016年4月に買収して、一気に知名度が上がりました。 このセッションでは、Xamarin とは、Xamarin.Android とは何か、C# や .NET Framework(Mono) の強力な言語・ライブラリ機能について触れ、通常の Android アプリ開発と Xamarin を使ったアプリ開発はどこが違うのか、どこが同じなのかを説明します。 また、今日のモバイルアプリ開発では、DataBinding、MVVM、Reactive Extensions(Rx) といった、Microsoft が源流となっている手法が広まって来ています。Xamarin を使うと、MVVMパターンと Rx を使用し、大部分のコードを共有できる Android/iOS 両対応アプリケーションを開発できます。如何にしてコードを共有するか、できない場合にどのような解決策が用意されているかについてお話します。 ■目次(仮) 1. Xamarin とは？ ・Xamarin とは何か ・Xamarin.Android とは何か ・C# の利点(Java, Swift との比較) 2. クロスプラットフォームアプリ開発とコードの共有 ・MVVMパターン ・Reactive Extensions / ReactiveProperty ・Portable Class Library(PCL)によるコード共有 ・プラットフォーム固有の処理を行う方法 3. Open Xamarin, Open Microsoft ・Xamarin で使えるライブラリ(C#, Java) ・All Xamarin SDKs are open source ・.NET Standard 4. Xamarin の使いどころ ・採用すべきケースとしないべきケース	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	amay077
2016/10/20 0:56:20	The Journey from Legacy Code to Clean Architecture	Buffer for Android was first made in 2012 - and since then it's seen a range of iterations and features. Whilst it's important for a product to grow in-order to meet the needs of users, it's easy for code to become unclean and lose any sense of architecture. This legacy code makes it difficult for us to maintain our product and build new features without the fear of breaking the app. But there is light! At Buffer we've just started to re-architect and refactor our application for Android using a clean architecture approach. Re-crafting our application feature by feature with the help of RxJava, Dagger and tests in the form of JUnit and Espresso has begun to result in a more stable, better architected and maintainable codebase. In this talk we will explore the journey we took through refactoring from legacy code to a clean architecture. Starting with heart of our app, the Composer, we'll share our learnings and findings discovered on our journey and the benefits from doing so.	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Joe Birch

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 23:05:04	minneにおけるAndroidチーム開発	<p>対象: 初心者～中級者 Android開発者が3人ほどのチームがスムーズに開発するために何をしているか知りたいたい人</p> <p>ハンドメイドマーケットminneのAndroidアプリ開発は、2年ほど前まで1人で開発されてきましたが、今は3人で開発しています。ここ最近では「複数のキャリア決済対応」「ハッシュタグ」「ゲスト購入」といった大きな機能をスケジュール通りまたはスケジュールよりも早く開発・リリースしてきました。</p> <p>これは、1人で開発していた時よりも安全かつ高速に開発するために、MVPアーキテクチャ導入を進めて単体テストが増えた事も大きく影響していますが、他にもスムーズに開発するためにAndroidチームとしてやってきた事がたくさんあります。</p> <p>今回は、Androidアプリ開発をしている3人が、開発をスムーズに行うためにやってきた事・やらなくなった事を説明していきます。</p> <ul style="list-style-type: none"> - 開発環境 - MVPアーキテクチャ - テストの日 - コードレビュー - プルリクエストイシューテンプレート - レビュータイム - 改善の時間 - リリース担当 - 朝会 - 褒める夕会 - Androidチームふりかえり - リマインダー - など 	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	hisaiichi5518
2016/10/29 17:17:54	Android Resources Refactoring	<p>Androidでは、resディレクトリ下で様々なリソースをxmlで管理しています。リソースはアプリの規模が大きくなるにつれて増えていき、秩序のない状態になります。開発しているアプリのcolors.xml, styles.xmlが綺麗に整理整頓されていると言えるの方が少ないのではないでしょうか。</p> <p>私の所属するQuipperでは、デザインの統一に合わせて無秩序だったリソースまわりを整理しました。デザインの變更にも強い設計で、エンジニアがデザインの実装で困ることも少なくなり、今見てもかなり綺麗に整理できていると思います。</p> <p>このセッションでは、colors.xml, dimens.xml, styles.xmlなどのリソースの要素の命名やファイルの分け方について私なりの理想の指針を説明します。その上で、既存のアプリのリソースをどうやってその理想の状態まで修正していったのか、チーム内でどうやって情報を共有していったのかといった、「リソースまわりのリファクタリングをどう進めていくか」という面白い知見もお話したいと思います。</p> <p>アプリのリソースを綺麗に保ちたい方、あるいは綺麗にしたいけれども効果や進め方がわからず足踏みしている方のお役に立てば幸いです。</p> <p>## 草案(仮)</p> <ul style="list-style-type: none"> - Quipperでのリソースまわりの指針(命名規則、ファイル分割規則) - styles.xmlは機能/画面単位で分割しよう - リファクタリングBefore/Afterのインパクト - リファクタリングのロードマップと締切の設定 - リファクタリングはcolors.xml, dimens.xmlから始めよう - チーム内でデザイン番長を決めよう - 規則をwikiに書いて共有しよう - 細かい修正タスクを洗い出して他のメンバーに練習させよう 	メンテナンス (Quality and Sustainability)		50分 / 50 minutes	konifar
2016/10/21 15:10:16	Embracing Layout Editor and ConstraintLayout.	<p>Since at the beginning of Android Development, ADT (Android Development Tools) provide us a Graphical Layout Editor where you can choose a widget and drag it into the editor and then edit the size or position of that widget as you like. Unfortunately, that Graphical Layout Editor didn't match the expectation. There were a lot of complains about the performance, the layout rendered in the Graphical Layout Editor and the result in real device are different. In the end people just gave up with the Graphical Layout Editor and choose to write XML directly instead.</p> <p>But recently Android Tools teams released Android Studio 2.2 that comes with new Graphical Layout Editor and Constraint Layout. The new Graphical Layout Editor is really powerful and easy to use, with a lot of feature like Blueprint mode and Properties panel for setting properties of that widget. With the new Graphical Layout Editor, Android teams also released a new Layout called Constraint Layout that will also solve an old Android UI problem UI nesting. In the past we need to put a LinearLayout inside a LinearLayout to achieve the expected UI, or we can also put the childView inside a RelativeLayout and set the anchor of every view to achieve the expected UI. With that approach (especially LinearLayout one) we will ended up with nested Layout that will make the drawing time longer and inefficient. With ConstraintLayout, you can create a flatter layout with defining constraints between the view or use the new Graphical Layout Editor's automatic constraint inference engine to create the constraints automatically and then you will get same result with flatter layout and shorter development time (especially with automatic constraint inference engine).</p>	UI-デザイン (UI and Design)		30分 / 30 minutes	Niko Yuwono
2016/10/21 17:49:29	イケてる継続的インテグレーション	<p>よく使われていた sdk-manager-plugin が deprecated になりました。けれどプラグインを使わずにうまく継続的インテグレーションできる方法があります！</p> <p>下記のネタをカバーします:</p> <ul style="list-style-type: none"> ・gradle プラグインの自動ダウンロードの使い方 / TIPS ・Jenkins / CircleCI / TravisCI の比較 ・Firebase Test Lab と CircleCI の連携方法 ・公開プロジェクトのキースタアの扱い方 <p>このセッションに参加された方は継続的インテグレーションを実現するのに必要な知識を持ち、最適なCI環境を構築できるようになるはず！</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	ケバド・ブノア
2016/10/21 19:09:04	Android OS 2.x時代の昔懐かしいアプリがマテリアルデザインを手に入れるまでのお話	<p>某インターネット企業(非スポンサーセッションのため非公開、問題なければ公開します)では近年、アプリに力を入れ始めました。そんな中、誰も正式な担当が付かずマテリアルの運用で維持されてきたAppウィジェットがありました。Android OS 4系への頃に作られたこのアプリは立体的で丸っこい見た目をしていた、見るからに懐かしさを感じさせる代物でした。</p> <p>2015年、ある取引先企業の意向により、この懐かしいデザインのAppウィジェットを、流行のマテリアルデザインに改修してくれたとの素敵な要望が届き、ボランティアとしてAppウィジェットの運用に携わっていた私は、デザイナーの方がいない状況にもかかわらず、この要望に応えるべく独学でマテリアルデザインを学び、業務の空き時間を利用して実装に取り組み、リリースまで持っていました。</p> <p>もちろん、Androidアプリはリリースしてからが本番です。ボランティア状態の開発者が1人しかいない状態でもクラッシュレポートは上がりますし、Android OSは進歩を続けていきます。それらの対応も当然必要となります。そして、お察しの通り、そのAppウィジェットはEclipse+ADTプラグインという皆様の記憶に深く刻まれているであろう環境で開発されていました。どういことかは皆様よくお分かりだと思います。</p> <p>このセッションでは、ただの苦勞話をするつもりなので、お聞きにいらした皆様の業務で役立つ知見を伝えられるかは自信がありません。最新の技術の話はまったく出てきません。イカした開発環境やテクニックの話も出てきません。それどころか環境を少しでも良くしたい、という強い気持ちをお持ちの方の助けになる話ができればと思っております。</p> <p>対象:</p> <ul style="list-style-type: none"> ・Android OS 4系以前の昔懐かしいアプリをそろそろMaterial design対応させたい、野心溢れるAndroidアプリ開発者 ・ゆでガエルになりにたくないAndroidアプリ開発者 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Hiroki Kitamura

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/21 23:24:50	Function introduction of Google Play Services	<p>■概要 Google Play Servicesには様々な機能がありますが、あまりよく知られていない機能も多くあります。その中から今回はGoogleサインインを簡単に実装するためのGoogle Sign-In API、Google+1ボタンを簡単に実装することができるPlusOneButton、メールやSMSで友人・知人を簡単に招待できるApp Invites、Google Play Servicesに依存しているFirebaseの一部の機能についての実装方法や実装してどう変わったのかなどの検証結果についてお話しします。</p> <p>■対象者 - Androidアプリ開発をある程度やったことのある人 - App Invitesに興味がある人 - Google Sign-In APIに興味がある人 - ASOが気になる人</p> <p>■目次案 - Google Sign-In API - 以前のGoogleサインイン - APIの概要 - このAPIの何が良いのか - 実装方法・サンプルコード - Google Plus One Button - Google+1ボタンとは - Google+1はGoogle Playの順位に影響するのか - やったこと(どう実装したのか) - 検証結果 - これはやるべきなのか - App Invites for Android - App Invitesとは - App Invitesのフロー・仕組み - Firebase Invitesと何が違うのか - App Invitesの数値のとり方 - 実際に実装して検証してみた結果 - これは導入すべきなのか</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	syarihu
2016/10/22 3:00:42	Practical Dagger 2 - Beyond Testing	<p>In the current climate of Android development many developers stress the importance of clean architecture and testing, myself included. But often it is a difficult sell to product/management (Or even technical directors) to allocate time to avoid technical debt and stop regressions. This talk will be go from setting up a project with an implementation of Dagger 2 to supercharging your development and removing the plumbing of your app that would otherwise by necessary without Dagger 2.</p> <p>The goal of this talk will be to take a basic understanding of Dagger 2 and to bring you closer to being able to implementing it into your app so it aides you regardless of whether or not you end up having time to test.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Patryk Poborca
2016/10/23 16:18:30	こわくない！ Fragment	<p>概要 好むと好まざるとにかかわらず、普段からFragmentを使う機会は多いとおもいます。しかし、ライフサイクルが複雑だったりハマりどころがあったりして苦手意識を持っている方も少なくないのではないのでしょうか。普段は使うばかりでなかなか実装を読む機会のない、Support LibraryのFragmentや関連クラスのソースコードを見ながら、Fragmentの仕組みを学び、Fragmentと上手に付き合えるようになることを目指します。</p> <p>対象 Android FrameworkやSupport Libraryにある程度慣れている中級者以上を対象とします。</p> <p>内容案 ・FragmentActivityはいかにしてFragmentを保持しているのか ・Argumentsは誰が管理しているのか？ ・Fragmentのライフサイクルをコードレベルで追いかける ・LayoutからinflateされたFragmentの扱い ・Fragmentの仲間たちとの付き合い方 - FragmentPagerAdapter系、DialogFragment系</p>	プラットフォーム (Platform)		30分 / 30 minutes	kikuchy
2016/10/24 4:14:10	Keeping it SOLID	<p>Fed up of code that's difficult to refactor? Tired of code smells? Tired of working with messy software architecture? Having personally worked with legacy code on a regular basis, I began to understand how important it is to allocate these pains. Luckily for you there are five principles of object-orientated design that can help us to craft software that is both easy to maintain and extend, these principles are known together as SOLID.</p> <p>In this talk we'll look at what makes up the SOLID acronym and what each principle means for our code - followed by how we can use these principles to improve both the code we maintain and also code we write in the future. Working with legacy code on the android app at Buffer has given me the opportunities to apply these principles when both refactoring and rewriting parts of the project. From this we'll be looking at real world examples to help us understand how the different aspects of SOLID can be implemented into our own projects to create the same result of a better crafting system.</p> <p>By the end of this talk, we'll be more familiar with the concepts of SOLID and how we can use the approach during development to write more maintainable and extensible code.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Joe Birch
2016/10/24 17:14:12	Doing Android Libraries The Right Way	<p>As more and more people start building Android Libraries, the whole process of building a better api for developers is getting bloated and everyone is coming up with their own ideas. However if android library developers follow a certain standard with a rationale they can avoid most of the pitfalls. Building an android library is easy, but building one that keeps in mind developer happiness is rare but definitely not hard, provided the android library developer keeps in mind a certain set of points such as every developer who uses the android libraries faces a common problem of inconsistent and non-informative api exposed by android sdks developer. All that can be avoided and fixed very easily during the time of planning and developing the android sdk. The session would cover how one can expose more informative api, make sure the best resources are utilized and that one does not force the app developer who is using our android library with redundant dependencies and complexities. In a gist, the session would be covering the do's and don'ts of building a proper android library.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Nishant Srivastava
2016/10/24 22:14:25	エンジニアが武器にする Material Design	<p>■概要 Material Designは2014年にGoogleが提案したデザインガイドラインであり、最近では多くのAndroidアプリで採用されています。 ノハナ社では2016年1月からエンジニア主導でアプリのMaterial Design化をじっくりと進めています。エンジニアがデザインに関わるとどのような利点があるのかをノハナ社の事例を元にお話しします。</p> <p>■目次案 - エンジニアがMaterial Designを理解する利点 - Material Designの特徴とエンジニアの思考 - デザイン巻き戻り工数の削減 - エンジニアにしかできないユーザー体験の向上 - Material Design化の進め方 - ノハナ社のデザインフロー - Material Designを広めるために - 社内勉強会とその後のフォロー - Material Designの思想背景を理解するための思考法 - なぜBottom navigationではBackキーでコンテンツを切り替えるべきでないのか - なぜSupport LibraryにもデザインガイドラインにもProgress Dialogがないのか</p> <p>■対象者 - Material Designをなんとなく知っていてもっと理解を深めたい方 - デザイナーの言いなりで実装するのが嫌な方 - その他、アプリのUIを改善したい気持ちを持っている方</p>	UI・デザイン (UI and Design)		50分 / 50 minutes	瀬戸優之

best	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/24 23:35:09	Data Bindingで実現するMVVM Architecture	<p>■概要 Data Bindingを使うことでAndroidでもMVVM Architectureが簡単に実現できるようになりました。View-ViewModel-Modelの責務を明確にすることで、見通しの良いコードが記述できるようになります。MVVMで実装する時のポイントなどについて話します。MVVMの導入を検討している方や、興味のある方の参考になれば幸いです。</p> <p>■内容案 * MVVM概要 * Data Binding概要 * Viewの役割、実装例 * ViewModelの役割、実装例 * RxJavaと組み合わせる</p> <p>■対象者 設計に興味がある方</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Kenji Abe
2016/10/25 9:56:09	Androidで音声認識を使いこなす	<p>■対象 Android アプリ開発経験者初級者～上級者 音声認識に興味があるが、触り程度しか使っていない方</p> <p>■内容 近年 Googleの「OK google」のCMや、iOSのSiriの普及により、スマートフォンで音声認識を行うことが一般的になってきた。これらの技術には音声認識技術が使われていることが明らかではあるが、文章入力なのか、コマンド入力なのか、学習データ入力なのか等用途によっても様々な手法を扱われる。本セッションでは、これらの様々な手法を解説するとともに、実際に自分で開発・実装するにはどうすれば良いのかを示す。</p> <p>■構成 * 音声認識の概要 * Android Speech Recognizerを使いこなす * 様々な音声認識エンジン * 連続音声認識とは * 自分のアプリで「OK Google」のような機能は作れるのか * 音声認識機能全般の評価について * Drivemodeでの活用</p> <p>■本セッションに関連する話題 * 音声認識 * 音声認識エンジン * 雑音抑圧 * Android * Service * プロセス間通信 * AIDL</p>	プラットフォーム (Platform)		30分 / 30 minutes	KAKKA
2016/10/25 10:43:26	RxJavaを使わないあなたのための、CursorLoader再考	<p>■内容 Activity/Fragmentのライフサイクルと同期し、標準のクラスとして提供されているCursorLoaderですが、最近RxJavaに奪われて影が薄くなりつつあります。CursorLoaderには、以下の欠点があります。 - ContentProviderを作らなくてはならない。 - 複数のCursorが必要な時には、その分だけCursorLoaderが必要。 - Cursorにしか使えない。</p> <p>本セッションでは、簡単なカスタムのAsyncTaskLoaderを導入することで、これらのCursorLoaderの欠点を全て解消した以下のLoaderを作る方法を提示します。 - ContentProviderを経由せずに、直接SQLiteにアクセスできる。 - 複数Cursorのみならず、任意のデータをロードできる。</p> <p>■対象者(中級者以上) - ContentProviderなんて簡単だと言われて作ってみたいが、JOINやprojection mapが必要になり苦しんだ人。 - RxJavaを使う世の中の風潮に抗いたい人。 - Androidの標準APIでがんばりたい人。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Hiroshi Sakurai
2016/10/25 13:06:22	Decoupled, Testable, Maintainable – Clean architecture for Android	<p>Building decoupled, testable and maintainable code for multiple devices is hard. Done right, it can tremendously increase the quality of your applications. This talk shows an architectural approach that allows building modular components for robust app development on all devices.</p> <p>The architecture presented, based on uncle Bob's "clean architecture", gives developers all the knowledge they need to create the essential layers of their applications. Small modular classes allow easy unit testing, while maintaining a clear separation of concerns allows having well defined business rules.</p> <p>This talk includes a demo, showing how the presented concepts can be applied to many common use cases: Database access, reading from and writing to the cloud and accessing platform dependent functionality, like the device's location.</p> <p>知ってます知ってます。いまの最新版 (droidkaigi開催当時) 7.1であることを。</p> <p>とはいえAndroid 7.0対応の前にAndroid 6.0の対応が出来てないまま、誤魔化し誤魔化しアップデイトを続けているアプリがあるんじゃないでしょうか？ いまはなんとかなっても、いつかはどうにもならず「苦渋の決断」をする日がくるのではないのでしょうか？</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Johannes Borchardt
2016/10/25 18:49:14	いまからはじめるAndroid 6.0対応 ~Android 7.0から8.xを見つめて~	<p>本セッションでは7.0に向けた復習として、自分が業務で「いやここで対応すべきだ」と決断していくつかのアプリを6.0対応した際に得たノウハウを共有いたします。さあ、みなさんも早いうちに6.0対応を果たし、気持ちよく7.0対応を迎えましょう！</p> <p>主に話す内容 * Apache停止に伴う対応 (さよならVolley, こんにちはokhttp) * ハードウェア対応と、コード以外に考えること * その他、Android 7.0に加え、さらにその先のアップデートに備えて気をつけることはなにか</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	yamacraft
2016/10/27 1:34:31	ドキ★脆弱性 onCreate() から onDestroy() まで	<p>■対象者 初心者～上級者。 Androidアプリで実際に公開された脆弱性について興味のある方。</p> <p>脆弱性はバグです。バグは必ず産まれるものです。例えセキュアコーディングガイドを隅から隅まで暗誦してたと、予期せず脆弱な実装は世にでるでしょう。本発表ではJVN49343562とJVN61297210が付与されたAndroidアプリ「マネーフォワード」の「WebView クラスに関する脆弱性」「任意の操作が実行可能な脆弱性」について、下記のアジェンダ(仮)でご紹介いたします。</p> <p>- 脆弱性対応タイムライン - 脆弱な実装の紹介 - 脆弱な実装の修正 - 脆弱な実装の背景 - 俺たちはどう脆弱性に向き合っていくのか</p> <p>こうしたノットベスト・プラクティスを共有することで、世の中のアプリの品質がより高くなることを願っています。又、一歩踏み込んでこういったバグをも知見としてオープンに共有できる文化作りを一端担いたいと思います。</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	ken5scal/鈴木研吾
2016/10/27 8:26:43	Keeping your app in shape	<p>Having a performant Android app is a requirement to keep users happy. This talk will be an exploration of different techniques and tools I have used to make apps better.</p>	メンテナンス (Quality and Sustainability)		50分 / 50 minutes	Carlos Sessa
2016/10/27 14:38:31	RxJava for Beginners: Benefits and Downsides	<p>There has been a lot of chatter about RxJava in Android lately. While some developers have fully embraced it, there are still some who have reservations. Due to the steep learning curve that comes with using RxJava, many have not had the time to fully understand its benefits and downsides. Is RxJava just another new shiny technology? Or can it actually help your development process? In this talk, we will look at the pros and cons of using RxJava. I will also show examples of common beginner mistakes when using RxJava. There will be code snippets to compare how your code can look with and without RxJava. We'll also look at common use cases of RxJava, and go through a few examples of when it can be helpful. Attendees will walk away with a better informed decision of how, if at all, RxJava can be used in their projects.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	Caren Chang

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/27 14:49:15	App Shortcuts in Android Nougat 7.1	In Android Nougat 7.1 the ability to create app shortcuts was introduced. Implementing shortcuts into your app helps guide users to specific parts of the app. If implemented correctly, it can help simplify the way users interact with your app, thus providing a delightful user interface experience. In this talk we will see an example of how to create app shortcuts. We will also talk about the different kinds of shortcuts you can create (static and dynamic) and how to chose which one to implement. Lastly, we'll talk about how to create a proper back stack of activities once the user launches the app through a shortcut. By the end of the talk attendees should have a clear understanding of how to help increase user engagement by making it easier for users to use their app.	UI-デザイン (UI and Design)		30分 / 30 minutes	Caren Chang
2016/10/27 16:23:44	How to stay up to date with the latest Android technology	I would guess this talk will be the most beneficial for those developers who are new or have 1-3 years of experience in Android development. Technology is evolving every single moment, even right now. Although it does not mean you have to be the first one to know the latest information, we, Android engineers, are all obligated to be up-to-date with the latest technology. This is because choosing and applying the latest technology can be (or can NOT be) the best solution for our essential daily task: create new Android apps or solve existing problems that your apps encounter in order to achieve the best outcome possible with the minimum costs required. This talk will show all the practices and attitude that I have personally practiced, enhanced, and grown over time to stay up to date with the latest Android technology, such as - Recommended external sources: websites, podcasts, conferences, etc. - Daily routine that I do every day for input - Attitude and opinion toward "output for input" By the end of this talk, attendees will know a list of external sources that they can start checking every day, sample routine work that may or may not fit to their daily lives, and one attitude or opinion toward input process through outputting the knowledge, all for staying up to date with the latest Android technology.	その他 (Other)		30分 / 30 minutes	Shohei Kawano
2016/10/27 19:19:11	Androidで使えるJava/C++相互運用環境'17	JavaとC/C++で書かれたネイティブライブラリの相互運用は面倒な問題です。Javaの世界では、この問題にさまざまな人たちが取り組み、数多くのバインディング自動化機構を構築してきました。本セッションでは、特にAndroidでの利用可能性を意識しながら、一方でJava以外の技術も視野に入れつつ、それらの技術のいくつか、実例も含めて紹介したいと思います。(この種のフレームワークはmoving targetなので、当日まで主にどれを紹介するかは未定です。)	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Atsushi Enno
2016/10/28 11:48:36	Firebaseを自社アプリに取り入れた話	【対象者】 自社サービスとしてアプリを運営している開発エンジニア、およびアプリ開発を受注しているSier関係者 【発表内容】 2016年5月にメジャーバージョンアップデートが行われたFirebaseを自社アプリに取り入れるにあたって、決断者で導入するためのアグリーを取りに行った話をします。「新しいことを始めたい!」「もっといいやり方を提案したい!」「でも、上司、お客さんは従来通りの古いやり方しか理解していないから提案できないという方に提案の仕方の一例としてご紹介できればと考えております。また、導入にあたって注意したポイントもご紹介いたします。	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	高澤 英悟
2016/10/28 17:28:38	Building my own debugging tool on overlay	■対象 アプリのデバッグに工夫を加えたい初級者~ ■内容 アプリの動作を検証する上で、画面の表示以外の目に見えない部分で何が起きているかを知ることにはとても重要です。デバッグ実行中であれば、デバッグをアタッチしてステップ実行をしたり、各モデルの状態を都度確認したりすることができますが、QAなどリリースビルドを用いた動作テストではデバッグを使うことができません。また ProGuard の設定によってはログ出力も削られてしまうため、動作に問題が起きたときの検証に工夫が必要となります。 本セッションでは、Android の開発者オプションで使われているような各種デバッグ情報の表示を参考に、デバッグに役立つであろうログ出力を画面に表示する方法を提示します。これによって、誰でも簡単に自分で同様のツールが作れるようになります。	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Keith Yokoma
2016/10/28 18:07:07	分野が違う人たちが作り上げるAndroidチーム開発	対象: 初心者~中級者、チーム開発に興味がある方 概要: このセッションでは、スタートアップの会社がAndroid開発を少ない人数でいかに爆速で開発したか、チーム開発や手法などについてお話しします。 私が所属しているSpacemarketではプラットフォームとしてWeb, iOSは既にありますが、Androidだけが遅れました。私とWebエンジニア、デザイナーの4人でアプリを作る上でチームとしてどうやって開発を進めたのか、アーキテクチャの選定、開発における苦悩など開発の裏側の泥臭い部分をお話したいと思います。	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Akitsugu Tamagawa
2016/10/28 18:41:38	Android High Performance	Performant applications are one of the key drivers of success in the mobile world. Users may abandon an app if it runs slowly. Learning how to build applications that balance speed and performance with functionality and UX can be a challenge; however, it's now more important than ever to get that balance right. This session will show you how to wring the most from any hardware your app is installed on, so you can increase your reach and engagement. An introduction to state-of-the-art Android techniques and the importance of performance in an Android application will be introduced, and later we will slowly walk through the Android SDK tools regularly used to debug and profile Android applications, learn about some advanced topics such as building layouts, multithreading, networking, security and battery life (one of the biggest bottleneck in applications).	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	Enrique López Mañs
2016/10/28 18:54:45	失敗から学ぶアプリを運用するために開発で考えるべきこと(ぼっちエンジニア編)	対象: 初心者~中級者のぼっちエンジニア 2011年頃から業務でAndroidアプリの開発を続けてほしい6年。 開発する上で「良かれ」と思った仕様や設計で失敗したこともたくさんありました。そしてこの6年、幸か不幸か複数のアプリを掛け持ちしながら運用を続けた経験もありました。 そんな中で痛感したことは「運用」を意識しないで開発すると、本当に「痛い目」に会うなという事。今回はそんな6年の中で感じた「痛い目」に触れながら「運用することを意識しながらアプリを開発するには何を考える必要があるか、そのためにエンジニアに何が必要か」を共有できればと思います。 こんな感じの人がこんな感じのことに触れるかもしれない一覧 - プロデューサー、サーバサイドエンジニア1名、アプリエンジニア1名、デザイナー1名体制だ - サーバサイドエンジニアとの兼がりがAPIしかない - プロデューサー含め開りがiOSユーザーばかりだ - アプリデザイナー経験がないデザイナーとのふれあい方 - 「運用」ってなんだ? - なにが「良いアプリ」なんだろう - 検証端末に肌えとしのぎ方 - 『自衛』のための知識吸収と「攻め」 - 「強い」エンジニアになる - 失敗したライブラリ選定、「安牌」ってなんだろう - 「必要な人間」になろうとしたいけない - お金、一番大事 内容にはかなり特殊/後ろ向きな内容が含まれる恐れがあります。	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	yamacraft

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/28 22:58:04	Practical Vector Drawable	<p>SupportLibrary23.2よりsupport-vector-drawableが追加され、Android L未満でもVectorDrawableが使えるようになりました。 VectorDrawableを使えば、画面解像度ごとの画像リソースを減らせるだけでなく、色や大きさの変更が容易になるなど様々な恩恵を受けられます。</p> <p>一方で、VectorDrawableは使い方を間違えると描画に時間がかかってしまったりするなど、注意しなければならぬ点もいくつかあります。 また、既存のpng画像をvector画像に置き換える場合、デザイナーとの協業やエンジニア同士のレビューも少しやり方を変える必要があります。</p> <p>本セッションでは、VectorDrawableの内部の仕組みから導入方法を軽く説明した後、効果的に使うための実践的なTipsをいくつか紹介します。 以下の内容に触れる予定です。</p> <ul style="list-style-type: none"> - VectorDrawable Internal - 既存のアイコンからVectorDrawableを作る - VectorDrawableの便利ツール紹介 - VectorDrawableアンチパターン - SketchファイルからVectorDrawableをエクスポートする部分の自動化 	UI-デザイン (UI and Design)		30分 / 30 minutes	konifar
2016/10/29 3:15:16	Beyond Multidex - Scaling Android cosebases	<p>When scaling an Android team from 15 to 30+ engineers in less than a year and in a codebase with over 20k methods, unforeseen problems start to arise in areas which are not usually seen for smaller teams and apps. For example, build times, developer productivity, IDE performance and a whole set of new challenges are faced every day, making it a non trivial task. In this talk, we'll cover essential tools and techniques used at Airbnb while scaling the team, while maintaining a fast pace and sanity.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	felipecsl
2016/11/01 7:01:45	Android CDD 7.0	<p>■対象 Android APIの仕様のあるアプリ開発者・フレームワーク開発者 Androidデバイスの仕様のあるアプリ開発者・フレームワーク開発者 (実装の技術レベルは問いません)</p> <p>■内容 CDDとは Compatible Definition Documentation の略で、Androidの互換性を定義した文書です。 2016/10/29に Android 7.0 Nougat に対応したCDD 7.0 が公開されました。 CDD 6.0は74ページの大作でしたが、CDD 7.0では85ページとさらにボリュームアップしています。 本セッションではCDD 6.0からCDD 7.0で変更及び追加された内容について解説します。</p> <p>■構成 * Android CDDとは * CDDとCTSの関係 * CDD 7.0の概要 * CDD 6.0からCDD 7.0の変更・追加された内容</p> <p>■本セッションに関連する話題 * Android Framework * 後方互換性 * 機種後換性 * SDK</p>	プラットフォーム (Platform)		30分 / 30 minutes	roishi2j
2016/10/29 12:55:51	Best Practice for Fragment	<p>Fragment利用時のお作法、はまりポイント、ベストプラクティスを総復習します。 Navigation Drawer や Bottom Navigation, ViewPagerと組み合わせる際に、メモリーリークを回避して正しく画面回転に対応する方法をご紹介します。</p> <p>スマートフォン向けWebサイトを作る際の、Android向けtips集です。</p>	プラットフォーム (Platform)		50分 / 50 minutes	あんざいゆき(yanzm)
2016/10/29 13:09:26	WebページのAndroid対応Tips	<ul style="list-style-type: none"> - theme-colorタグでシステムUIとの親和性を高める - ショートカットをホームに追加 - バックキーでモーダルを閉じよう - @を絵文字にしないために - その他... 	その他 (Other)	Web	30分 / 30 minutes	mstssk
2016/10/29 13:13:53	トークアプリで絵文字を実装した話	<p>対象: Android開発の経験がある方、興味がある方</p> <p>概要: 業務で開発していたトーク機能を持つアプリに独自絵文字を実装した話をします。</p> <p>誰かとメッセージのやりとりをする際に、単純な文字だけ送るよりも絵文字を追加することで表現豊かなメッセージを送ることができます。 絵文字はガジェットの時代から存在しており、スマホがメインになった現在においてもキャリア独自の絵文字やキーボードアプリ独自の絵文字など様々な絵文字が存在しています。 私が開発していたトーク機能を持つアプリでもそういった絵文字が多く使われていたのですが、端末の違いによって送信側と受信側で異なる表示になってしまうことも少なくありませんでした。</p> <p>そこで、アプリ独自絵文字を実装することで絵文字を使った表現豊かなメッセージを、表示の違いなどが起きないように可能にしました。 実装にあたってはトークで使うため表示の高速化や絵文字自体のクオリティにも注意して実装していました。 発表では実装にあたって苦労した点、どうやって乗り越えたか、そこから得た知見などお話できればと思います。</p> <p>目次(予定): - 絵文字導入にいたる経緯 - 絵文字導入前に必要な機能を洗い出す - 実装中にあった壁と解決方法</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	futaboo
2016/10/29 13:17:46	How to apply DDD to Android Application Development	<p>MVPだけのクリーンアーキテクチャのさまざまなアーキテクチャをAndroidに適用してみた話が花盛りですが、他のプラットフォームでうまくいったアーキテクチャをAndroidに適用してうまくいくでしょうか？アーキテクチャが何を目的としたものなのか正しく理解せず、技術的なパターンだけ適用してうまくいくでしょうか？</p> <p>このセッションでは特定のアーキテクチャではなく、ソフトウェア開発手法・設計理論であるDDD (Domain Driven Design : ドメイン駆動設計)をAndroidアプリ開発に取り入れる方法について話します。 DDDの内容については「エリック・エヴァンスのドメイン駆動設計」及び「実践ドメイン駆動設計」に準拠します。</p> <p>対象者 - DDDについて正しく理解したい人 - 特定のアーキテクチャの技術的なパターンだけを適用することに疑問を感じる人</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	あんざいゆき(yanzm)

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/29 13:56:49	コマンドなしでぼくはAndroid開発できない話	<p>■対象者 Android開発をする上で便利なコマンドを知りたい人 Android開発をより効率的に行いたい人 GUIよりコマンド(CUI)が好き人</p> <p>■概要 このセッションでは30分にかくAndroidに関するコマンドの話します。</p> <p>Android Studioでlogcatを見れたり、Android Monitorから色々見れたりしますが「それターミナルからコマンドで見れますよ」と言いたいわけです。 コマンドを介せばさらに細かく端末の情報を確認することができます。 コマンドから文字を入力したり、Backキーを押したりコマンドを介して端末操作もできます。 Intentの発行など人力で行うと面倒な作業もコマンドからなら楽にできます。 色んなコマンドを知ること、Androidの状態や仕組みを理解する機会にもなります。</p> <p>そんな「コマンドでAndroid開発を楽しみたい・効率的に開発したい！」というみなさまの熱い期待にお答えするセッションです。</p> <p>私が日々使っているコマンドからいざって時に役立つかもしれないコマンド等、色んなコマンドをご紹介します。実際にコマンドを実行しながらどんなことができるものなのかを見ながら紹介していくものも用意する予定です。 コマンドを知ることで、今まで以上に作業効率の向上、Android楽しさ・奥深さを知っていただけたら嬉しいです。</p> <p>具体的には以下の様なものをメインで話す予定です。 * adbコマンド * Androidのshellコマンド * adbを便利にするTool * Androidのコマンドを支える技術 * コマンドを組み合わせたTips</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	operandoOS
2016/10/29 15:01:47	N is for Notification	<p>本セッションでは最新の情報に基づいたノーティフィケーション(通知)機能を紹介します。 またAndroid OSのバージョン依存を解決するアプローチを理解できます。</p> <p>Androidの通知バーは最新の7.xに至るまでのあいだに多くの機能変更、追加が行われてきました。たとえばカスタマイズできるスタイル、Heads-up、グループ化などが追加されています。単純な通知機能以外にもタイトルAPIなどもあり、バージョン差が大きい機能といえます。</p> <p>本セッションでは、特にノーティフィケーションに焦点をあて、ユーザーが欲しい情報を効果的に届ける方法を紹介し、また機能だけにとどまらず、バージョンによる差分をどのように吸収するかというアプリ開発の課題についても触れていきます。 セッションではNotificationCompatを題材に機種依存、バージョン依存の攻略手法を例示しながらアプリケーション開発でバージョン差を吸収するための指針を示します。 Androidの歴史を振り返り、プラットフォームへの知識を深めるのに最適です。</p>	プラットフォーム (Platform)		30分 / 30 minutes	mhidaka
2016/10/29 15:07:15	How to remodel current testing environment	<p>【対象者】 ・新しく入ったプロジェクトのテストをどうにかしたい人 ・現状のテストの信頼性に不安のある人 ・テストを書くときどこまで書けばいいのかわからない人</p> <p>【内容】 BDDを始めとする思想では、テストを用いてそのアプリケーションの動作を保証・信頼します。テストがないプロジェクトは信頼できない、という話を聞いたりしますが、では「存在するだけで信頼できるもの」でしょうか。 あるいは「信頼できるテスト」は開発者にとって常に「良い物」なのでしょう。 各開発者が持つテストに対する意識が異なる場合、テストの粒度はまちまちとなり、信頼できるテストとできないテストが混在するでしょう。 またどんなにテストを信頼していたとしても、記法・表現方法などの問題からテストを書くこと自体が負担となることもあります。</p> <p>途中参加した2つのプロジェクトに対して導入したテストドキュメントの整備方針などから、具体例を交えつつ、上記の問題点をどのように解決するかをお話できればと思います。</p> <p>コンテンツ ・テストストラテジーの作成 ・テストポリシーを用いた開発者間の意識共通化 ・Test doubleという考え方のおさらい ・Test Size ・設計思想に合わせたテスト作成ポリシー ・MockableなAPI</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	red_fat_daruma
2016/10/29 15:23:27	AndroidTV「もしかして...」FireTV「俺たち...」「全然対応されてない〜!!」??」	<p>概要: TVデバイスは前前前世からブラウン管→液晶→3Dと渡り歩き、スマートテレビに辿り着きました。 近年、本格的にAndroidを搭載したTVデバイスが普及し始めています。 みなさんのアプリはTVデバイスに対応していますか? 私が担当しているプロジェクトでは先日FireTV/AndroidTV(仮)に対応しました。 しかし、現状TVデバイスに対応するための知見はまだ少なく、スマートフォン向けのアプリ対応と比べて開発の敷居が非常に高いと感じています。</p> <p>このセッションでは、今後TVデバイス向けアプリ開発を検討している人にフォーカスし、導入→開発→運用と実践的な話をします。</p> <p>主なキーワード: FireTV, AndroidTV, Leanback</p> <p>対象ユーザー: ・スマートフォン向けAndroidアプリ開発経験がある方</p>	ハードウェア (Hardware)		30分 / 30 minutes	ogaclejapan
2016/10/29 15:33:09	Androidアプリのストレージ戦略 ~AndroidってSDカード使えるんでしょ?~	<p>本セッションでは「AndroidってSDカード使えるんでしょ?」と急に出てきたちよつとヤバそうな仕様との付き合い方、「気づいたらアプリで容量使いすぎて死にそう」という機能要件をクリアするためのストレージ戦略を解説します。 内部/拡張などストレージの種類を問わず、アプリがデータを保持するための最適な場所について横断的に考察します。</p> <p>今は昔、古来のAndroidアプリではSDカードのパスを探すだけで一苦労がありました。現在、Android SDKが提供するExternal Storage APIでは拡張ストレージへのアクセスを可能にしており、アプリはより多くの情報を拡張領域に保存できます。</p> <p>セッションではアプリケーションの種類や保存するコンテンツの性質を考慮しながら内部/拡張ストレージを上手に使い分ける方法を紹介し、利用にあたってはマルチアカウント、端末の空き容量、クラウド連携、暗号化など設計上考慮すべきポイントを押さえます。 過去のTipsが今も使えるのか?という疑問やAndroidアプリ開発のスタンダードは存在するのか?という設計上の課題を解消します。</p>	ハードウェア (Hardware)		50分 / 50 minutes	mhidaka

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/30 17:13:22	実践 Kotlin on Android 2017	<p>DroidKaigi2017が開催される頃にはKotlin 1.0がリリースされて1年が経過します。本セッションでは2016年4月にKotlin 1.0をAndroidのプロジェクトに導入し、現在に至るまでに開発の中で得たノウハウについて解説します。</p> <p>Android開発に導入する際に懸念となるランタイムサイズやメソッド数に関する最新の情報、Kotlinが解決するAndroidアプリケーション開発の諸課題、Kotlinがアプリケーション設計に与える影響、kaptなどAndroidアプリケーションで利用する際の既存の課題などを中心に話します。次期バージョンであるKotlin 1.1や、Kotlinが今後どのように進化するかを議論するKEEPなどの情報を交えて、KotlinでのAndroidアプリケーションがこれからどのように変化しそうかなどについても言及します。</p> <p>目次案</p> <ul style="list-style-type: none"> - Kotlinの概要 - Kotlinの現状とマイルストーン - Kotlinが解決するAndroidアプリケーション開発の諸課題 - Kotlin導入に当たって検討すること - Kotlin on Android実例集 - Kotlinがアプリケーション設計に与える影響 - ここが難しいKotlinとXX (DataBindingやkaptなど) - KotlinとAndroidの今後 	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	sys1yagi
2016/10/29 16:40:47	マッチョActivityを改善した話	<p>マッチョActivityを改善した話</p> <p>pairs(https://play.google.com/store/apps/details?id=jp.eure.android.pairs&hl=ja)でマッチョActivityを改善した話をします。</p> <p>ざっくりと、pairsは以下のような不健康な状態でした。</p> <ul style="list-style-type: none"> - 1000行超えActivity/Fragment - Viewのコードとビジネスロジックが交じり合うActivity/Fragment - staticメソッド群で実装されたApiClient/DataAccessObject <p>このような状況で様々な問題に悩まされていました。</p> <ul style="list-style-type: none"> - どこになにが書いてあるのかわからない - 手を加えたら思わぬところに影響がでた <p>*本発表ではアーキテクチャに関する詳細な解説等は行いません。マッチョActivityを潰すためにどのように手を付けていったのか、進めていったのかをメインに話します。</p> <p>目次(予定)</p> <ul style="list-style-type: none"> - pairsについて - 改善前のpairs - 改善の準備 - ActiveAndroid->Orma/Retrofit 1->2 - 改善 - Model - View 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	@lvla0805
2016/10/29 18:50:44	位置情報を正確にトラッキングする技術	<p>Pokemon GOによって再び位置情報を使ったアプリに注目が集まっていますが、AndroidのLocation Managerをそのまま使っただけでは精度の良い位置情報をトラッキングすることはできません。このセッションでは</p> <ul style="list-style-type: none"> - バックグラウンドで安定的に位置情報を取得するアーキテクチャの作り方 - LocationManagerに設定する最適なCriteriaの作り方 - 高精度の位置情報を取得するための各種フィルターの作り方 <p>の順に、UberやNikeと同等かそれ以上の精度の位置情報トラッキングを可能にする方法を説明します。</p> <p>デモアプリと実際のフィールドテストのデータを用い、トラッキングアルゴリズム開発の過程でパフォーマンスをどのように検証していたかという方法論も説明し、参加していただいた方が自分のアプリのニーズにあった位置情報トラッキングアルゴリズムへカスタマイズしていくためのヒントも提供できればと思います。</p> <p>また位置情報精度とバッテリー消費量の関係についても実際の検証データを使って考察する時間を取りたいと思います。</p>	プラットフォーム (Platform)		30分 / 30 minutes	水鳥敬満
2016/10/29 19:40:06	スタートアップにおけるデータドリブンな意思決定	<p>■ 内容</p> <p>アプリの成長に重要な、データドリブンな意思決定について話します。</p> <p>一年強、スタートアップでAndroidアプリを成長させるための開発に取り組んできました。我々のようなアーリーステージのアプリでは、自分たちが提供できる価値を素早く見定めることが大切です。</p> <p>しかし当初、あまり重要でない機能に時間をかけてしまったり、余分な機能を盛り込みすぎてメンテナンスコストが増大してしまったりがありました。</p> <p>それを改善するためにデータを中心とした意思決定に力をいれたところ、より方向性が明確になり、自信をもった意思決定ができるようになってきたと感じています。</p> <p>同様の課題を抱えるサービスで役に立てて貰えるように、これまで得た知見を共有します。目標設定など計画段階の話から、自社フレームワーク実装まで、開発フローを回すための全体像を話す予定です。</p> <p>■ 構成</p> <ul style="list-style-type: none"> - なぜデータドリブンな意思決定が重要か - アプリの成長におけるKPI設定について - 弊社アプリでの具体的な業務フロー - 使用しているフレームワーク・ツール - その他工夫している点 など 	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Seiji Kasahara
2016/10/29 20:56:22	Androidエンジニアでもリモートワークがしたい!	<ul style="list-style-type: none"> - 対象者 リモートワークでの働き方を検討している企業やAndroidエンジニア また、災害時や緊急対応等の一時的なリモート環境を構築したい方 <p>■ 概要</p> <p>近年リモートワークという働き方が日本でも広まっていますが、Android開発の場合、どのようにテストを行うのか、端末の管理はどうするのか等様々な制約があり、なかなかリモートワークをしたくても出来ないという方が多いと思います。</p> <p>そこで実際に業務としてAndroid開発をリモートワークで行った際の、問題点やナレッジを話したいと思っています。</p> <p>■ 内容案</p> <ul style="list-style-type: none"> * リモートワークについて * リモートワークの利点 * Android開発のリモートワークとWeb系開発のリモートワークの差異 * Android開発におけるリモートワークの問題点 * リソースや成果物の共有方法 * 問題点の解決方法、及び解決方法選択に至った理由 * 導入したツール * Andoridエンジニアとしてリモートワークを行ってみての感想、及び所感 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	endoidou
2016/10/30 16:44:35	Kotlin 1.1 async/await and RxJava	<p>Kotlin 1.1で追加される言語機能は沢山ありますが、その中でも特に大きなものがコルーチンの実装として追加されるasync/awaitとジェネレータです。</p> <p>Kotlin 1.1ではコンパイラへコルーチンのための予約語が追加されます。async/awaitとジェネレータの実装はライブラリで提供されます。</p> <p>本セッションでは、コルーチンの概要、Kotlin 1.1でのasync/awaitとジェネレータの仕組みについて解説し、Androidアプリケーションでどのように利用すると良いかを紹介いたします。またasync/awaitとRxJavaの関わりについても説明します。</p> <p>目次案</p> <ul style="list-style-type: none"> - Kotlinとはなにか - Kotlin 1.0.xと1.1の差分について - コルーチンとはなにか - Kotlin 1.1におけるコルーチン実装の仕組み - ジェネレータの実装の概要と実例 - async/awaitライブラリの実装の概要 - asyncRxを読み解く - AndroidアプリケーションでのasyncRxの適用 - RxJavaとasyncRxの協調と設計 (Operatorを置き換えよう) - おまけ: coroutineとsuspendを使った継続の実装 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	sys1yagi

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/30 17:03:46	解剖 Kotlin ~バイトコードを読み解く~	<p>Kotlinの特微的な機能としてnull許容型や拡張関数などがよく取り沙汰されます。null安全だと高階関数で簡潔に記述したか拡張関数便利最高といった話は枚挙に暇がありません。確かに言語機能としていろいろとJavaに比べて便利なのは分かるんですけど、でもまあ別にJavaを使っていてクリティカルに困っているわけではないしな~学習コストとかチームへの導入コストを考えるとそこまで興味を感じられるわけでもないしな~みたいなそんな気分。わかります。</p> <p>本セッションではKotlinのコードをコンパイルして得られるJavaバイトコードを、可読化したりデコンパイルする事によって、Kotlinの特微的な言語機能がJavaでどのように表現されているかを読み解いていきます。これによりKotlinが肩代わりしてくれるボイラープレートコード群を明らかにします。Kotlinを使うことで省略できたボイラープレートコードが可読性をどのように高め、設計に影響を与えるのかについても言及します。</p> <p>目次案 - Kotlinの概要 - 本セッションのアプローチ - null許容型の正体 - 関数型とラムダ式の正体 - インライン関数の正体 - 拡張関数の正体 - データクラスの正体 - プロパティの正体 - デリゲートプロパティの正体</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	sys1yagi
2016/10/30 17:25:13	インスペクションとAndroid Lint Custom Ruleによる、単一責任実装の実践	<p>■対象者 初級者~中級者 ・メソッドが複雑にならないよう実践したい方 ・Android Lint の Custom Rule と静的コード解析について理解したい方</p> <p>■概要 TDDやDDDが嬉しい昨今、コード実装の「単一責任の原則(SRP)」意識⇒シンプル化が求められています。</p> <p>Android Studio には、インスペクションという強力なソース解析(指摘)機能があることをご存知かと思いますが、問題点のある実装部をエディタ上でハイライト表示したり、「Analyze」メニューの「Inspect Code...」により、指摘一覧を「Inspectionツールウィンドウ」でカテゴリ別にリストアップしてくれくれます。独自の単一責任チェックのインスペクションが欲しいと思われているいませんか。</p> <p>実は、Android Studio のインスペクションには、Android Lint も利用されているのです。</p> <p>Android Lint は、独自の Custom Rule を作成することができます。そして Custom Rule を作るためにAST(抽象構文木)を使ったJavaソースコードの静的解析機能も提供されています。</p> <p>この静的解析機能を利用した、単一責任となるシンプルな実装パターンを強制するオリジナルのAndroid Lint Custom Rule の作成(方法)を中心にお話を進め、Android Studio での単一責任実装の実践(利用方法)について発表したいと思います。</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	robo
2016/10/30 18:22:15	minneにおけるテスト~リリース~リリース後に行っている事の紹介	<p>■対象者 初級~中級者 テストやリリース周りの仕組みを整備していきたいと考えている人 リリース後のクラッシュ対応、レビュー対応について検討している人</p> <p>■概要 僕が所属しているminneではAndroidチーム3人で開発しています。複数人数で安定的にアプリをリリースしていくには、テスト~リリースまでの仕組みを整える事が必要になってきます。また、レビューの良い評価を維持していくために、レビューの対応についても実際にやってきた事を含めながらお話ししていきます。</p> <p>具体的には、以下の内容を発表します。</p> <ul style="list-style-type: none"> - リリースフローにテストをどう組み込んでいるか。 - テストの種類 <ul style="list-style-type: none"> - 自動的テスト <ul style="list-style-type: none"> - Unit test - UI test - CI - 手動テスト <ul style="list-style-type: none"> - 手作業による検証 - 構築しているCI環境の仕組み紹介 <ul style="list-style-type: none"> - Drone.io - DeviceFarm - Slack - リリース前の検証について <ul style="list-style-type: none"> - リリース担当 - 検証シート作成 - DeployGate - リリース <ul style="list-style-type: none"> - 段階的公開 - Crashlytics - リリース後 <ul style="list-style-type: none"> - クラッシュ監視 - レビューの監視 - レビュー返信の体制 	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	mapyo
2016/10/30 21:33:52	Androidアプリ開発の体力づくり	<p>■対象者 Android開発をまだやったことない or はじめたばかりの人 Android開発の今を知りたい人 Android開発 やっせいくぞ という気持ちの人</p> <p>■概要 Android開発の現状は日々変化し続けています。OS、デザイン、開発ツール、ライブラリ、テスト、設計など日々小さくても様々な変更があり、色んな情報が飛び交います。「Android開発難しそう」と感じたり、「やってみただけ何から始めたらいいのかわからない」みたいな経験をした方もいるかと思います。しかし、大事なポイントを少しずつ抑えていくことによって、Androidを継続的に学べるようになります。</p> <p>本セッションでは、私が過去・現在経験してきたことも交えながら、Android開発をやっていく上で大事なポイントをお話したいと思います。何からはじめたらいいのか、継続的に学ぶために何をしたらいいのかなどを話します。次に、実際にアプリを開発することを想定して、どのようなAndroidの技術が必要なのかなどを話します。</p> <p>最後に、Android開発の現状や新しい情報をどのようにキャッチアップしていくかなどを話します。※話す内容が幅広いジャンルを扱う可能性があるため、具体的な実装やコードは少なめになる予定です</p> <p>内容案: * どのようにAndroidを学ぶのか * 作るアプリによってどのようなAndroidの技術が必要なのか * 新しい情報をどのようにキャッチアップしていくかな * 開発に役に立つ情報はどこで手に入れるか * Android開発の今とこれから</p>	その他 (Other)	初心者向け・入門	50分 / 50 minutes	operandoOS

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/30 22:13:33	ワンソース・複数アプリの実際	<p>Quipper社では、グローバル向けの Quipperアプリと日本国内向けの StudySapuriアプリの両方を一つのソースからビルドしています。</p> <p>一つのソースにしておくことには、メンテが楽だという明確な利点がある一方、異なるプロダクトの要求をどうまとめるという問題があります。</p> <p>そこについて、開発時のノウハウ及び、ビジネス要求側まで選ってどうやって対処していったかをはなたいと思います。</p> <ul style="list-style-type: none"> - 内容案 - そもそもワンソースでできるプロダクトとは - ビジネス側の理解 - 開発者の意識 - BuildFlavor - ブランチ戦略 - FeatureFlag - リソースの管理 - 無駄なライブラリの削減 - リリースと開発の同居 - CI環境 	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	dagezi
2016/10/30 22:54:06	Systemアプリ開発入門	<p>■対象者 SystemアプリおよびAndroid Frameworkの開発者 初級～中級向け</p> <p>■概要 Systemアプリ開発入門 一般的なアプリと同じく、端末にプリインストールされるアプリについてもOSのアップデートと共にセキュリティ観点のアップデートが多く含まれます。プリインなら何が可能なのか、何が不可能なのか、一般的なアプリと違いどんなことに気を付けて開発をしなければいけないのか、その開発手法についてのノウハウをお話します。</p> <ul style="list-style-type: none"> - Systemアプリ開発入門 - Systemアプリ概要 - 一般的なアプリとの違い、権限の違い - 権限とSystemアプリ - HideAPI, SELinux, Platform署名 - Device Owner - Nugatでどう変わった？ - マルチユーザーとPlatformアプリ - マルチユーザーに対応するために - ユーザー間で共有したい情報を保持するために - Debug手法 - Using AndroidStudio - BreakPoint - Test - (おまけ) Systemアプリ開発でもいろいろしたい - 著名Library使用(APT)などなど <p>■キーワード プリインアプリ、Platform署名、Framework、SELinux、RuntimePermission、CTS、マルチユーザー</p>	プラットフォーム (Platform)		30分 / 30 minutes	kobashin
2016/10/31 2:33:56	Intro to Static Analysis: Your Dream Side-Job Is Here	<p>A Bug bounty Program, a crowdsourced based rewarding program for discovering a software bugs, is now get familiar in many industry. Offering \$8,000 at max, Google, of course encourages the program so does Line. Even United Airlines is conducting it.</p> <p>I personally believe us, Android developers have a great advantage in term of Android App Bug Bounty Program. After all we develop the app on daily basis. But HOW should we TEST apps developed by others? STATISTIC ANALYSIS is one way to go.</p> <p>In this session I will discuss, Process of Bug Finding(in theory) Static Analysis Methodology Static Analysis Tools Demo List of available Bug Bounty</p> <p>In the end, I hope listeners find a good entry point for side-job, and ultimately, Android developers can complement each other app to make all secure and safe.</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	Kengo Suzuki
2016/10/31 5:52:36	今更聞けないGradle	<p>突然ですがあなたのbuild.gradle、ちゃんと他人に説明できますか？ いつの間にかコピペの山になっていませんか？</p> <p>本セッションでは、Android開発を始めて比較的天が浅い方を対象にGradleの基本的な要素から大きく実践できるtips、pluginの作成方法や発展的な内容をサンプルコードと共に紹介します。</p> <p>ビルド周りは普段中々時間を割くことのできない分野だと思いますが、更なる生産性向上の為にこのセッションがお役に立てば幸いです。</p> <p>対象者: * Android開発初級者～中級者</p> <p>コンテンツ(仮): * Gradleについて(5min) * Gradleとは * Android Gradle Pluginとは(Android Studioとの連携) * Android開発における基本的な設定項目 * Tips紹介(20min) * Build variantsの活用 * Build typeとProduct Flavors * Merging resources, Java sources, manifests * Flavor Dimensions * Custom tasks * Custom taskの作成 * ビルドプロセスへの組み込み、除外 * sourceSetsを活用した効率的なtaskの記述 * Testing * Testing support libraryを利用したUnit testの記述 * Espressoを利用したinstrumentation testの記述 * test reportやJacocoを利用したカバレッジの取得 * Performance * Daemonや並列化、JVMパラメータを利用した高速化 * dexOptionを利用した最適化 * CircleCIやJenkins上でビルドを速くする工夫 * Gradle plugin(10min) * Gradle pluginの作成 * 内部アーキテクチャの簡単な紹介 * pluginの作成方法 * jcenterへの公開作業 * 開発を効率化するおすすめpluginの紹介 * 発展編(5min) * Kotlin gradle pluginの可能性 * BabelやBack,Kobaltとの比較 * Q&A(5min)</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	hotchemi
2016/10/31 7:49:30	Optimizing & Profiling UI Performance	<p>Performance matters if you want to be able to provide buttery smooth, exceptional user experiences. However, it can sometimes be difficult to track down and fix these issues. In this talk you'll learn what causes jank (skipped frames) to occur while animating such as scrolling, tips on how to avoid it in the first place, and how to profile for problem areas if it does happen to you.</p> <p>You'll learn how to use tools such as Hierarchy Viewer, Debug GPU Overdraw, Systrace, new in Android 7.0 Frame Metrics API, and others.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	Brenda Cook (kenodoggy)

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 8:16:39	What's New in RxJava 2.0	<p>いま RxJava 1.0 を使っている方を対象に、RxJava 2.0 の変更の背景や RxJava 2.0 での変更点、どのようなことに注意すべきかを説明します。</p> <p>RxJava 2.0 は 2016 年の 10/29 に公開された、RxJava の新しいメジャーバージョンです。1.0 に比較すると以下のような変更が入っています。</p> <ul style="list-style-type: none"> - Java 9 で標準となる予定の Reactive Streams 対応 - 分りにくかったメソッド名やクラス名の変更 - いくつかの API の挙動の破壊的な変更 <p>RxJava 2.0 では、RxJava 1.0 で実現できたことは、ほとんどそのまま実現できます。一方で、RxJava 2.0 などの書き方をするので、より Rx や Reactive Streams の恩恵に与えられるようになっていきます。本セッションでは、RxJava 1.0 でいま Android アプリを書いている方向けに、2.0 での変更点やどのように 2.0 の機能を利用するのが良いかという話を述べます。特に 2.0 での変更の背景となった Reactive Streams がどういうものかというところから解説することで 2.0 の目指す姿を明らかにしたいと考えています。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	hydrakecat
2016/10/31 8:57:52	少し幸せになる技術	<p>[対象者] 初心者～中級者向け</p> <p>[内容] Androidを開発、運用していく中で起こるつまづきやすい問題の解決、ちょっとしたことで開発が幸せになるテクニックを紹介。本質的な作業に集中できる時間を多く取れるようにしたいと思っています。</p> <p>以下は仮予定の内容です。</p> <p>問題</p> <ul style="list-style-type: none"> - ProGuard問題 <ul style="list-style-type: none"> - なぜProGuardで、ビルドができなくなるのか - メソッド数O4K問題 <ul style="list-style-type: none"> - 64k問題とは - メソッド数を知る - メソッド数を削減 - 限界突破の方法 <p>便利</p> <ul style="list-style-type: none"> - Android Studio 2.2 の機能 - リファクタリングで役立つショートカット - Google Play Console 関連 <ul style="list-style-type: none"> - 通知を受け取る - クラッシュレポートの雑談の解除 - Grade の設定 - キャッシュ周り - 署名の設定 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	kamedon
2016/10/31 9:23:15	A New View: Layout Editor + ConstraintLayout	<p>The new ConstraintLayout is not just a new component but a new method of building UI in Android. Coupled with the new and improved layout tools in Android Studio, the ConstraintLayout looks to provide developers with flexibility, adaptability, and efficiency, both in the layouts themselves and in the development process.</p> <p>This session will explore these new tools for building complex and efficient layouts. First, we will take a quick look at the history of the ConstraintLayout. We will then discuss why the Layout Editor complements building UIs with the ConstraintLayout and go through several examples of how to utilize both the Layout Editor's tools and the ConstraintLayout's features.</p>	UI-デザイン (UI and Design)		50分 / 50 minutes	Huyen Tue Dao
2016/10/31 9:24:15	From Java to Kotlin in production	<p>There are many talks about what Kotlin is, why you should use it and how to get started. If you know those things, this talk is for you.</p> <p>I am going to cover how we use Kotlin in a real-world large code base. How we converted from a pure Java App over to an (almost) pure Kotlin one.</p> <p>Along the way I will cover pain points. Where did it blow up? Where did we mess up? What bad practices did we make? How do we keep builds fast? Why kapt 2 saved us and more!</p> <p>Beyond the issues, we will also cover the more advanced parts of the language which lets us do things which were previously more difficult in Java. E.g. Pure Kotlin DI, Sealed Class State Machines, Fluid testing and more.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	chrisjnx
2016/10/31 9:31:52	実際、Wear2.0って、どうなの??	<p>Android Wear2.0では、スタンドアローンやキーボードなど、多くの機能が追加されています。Wearアプリを1系の時に作成し、2016年3月から運用しております。現在、2.0に対応させるために、実装と検証をしています。「運用してみようだったか!や2系に対応させたときの気づき」などを入れつつ、Wear2.0について話せたら良いと思っています。また、iOSとの連携を試したことも話せたらと思います。(実装は、今のところ見送ってます)</p> <p>対象 Android Wearに興味がある人</p> <p>内容案</p> <ul style="list-style-type: none"> - Wear2.0で新しく追加された機能紹介(1系と2系の違いも踏まえながら...) - 2.0で追加されたUI(キーボードなど)を使用する際に注意したこと、導入を見送ったこと - iOS連携を試そうとした結果 - 約1年くらいWearアプリを運用してきて思ったこと - Wearアプリを作る時に注意した方がいいところ 	ハードウェア (Hardware)		30分 / 30 minutes	Kohei Niimi
2016/10/31 9:37:08	LayoutInflator - friend or foe?	<p>When building Calligraphy (https://github.com/chrisjnx/Calligraphy) it uncovered some interesting aspects of the LayoutInflator and some unknown side effects.</p> <p>Calligraphy could be described as a partial "hack" - I think it's important to show to how this works and although a "hack", it's been done with safety in mind.</p> <p>I will cover how the LayoutInflator works - being one of the most important parts of the Android framework, Android Developers should understand what goes on under behind their Layouts.</p> <p>Secondary let's look at how Calligraphy hacks the lifecycle to allow you to inject into views at inflation time and then how you can roll your own version of Calligraphy!</p>	プラットフォーム (Platform)		30分 / 30 minutes	chrisjnx
2016/10/31 13:27:37	初心者向けライブコーディング	<p>人がコードを書いているのを見るのはとても勉強になります。知らなかった書き方だけでなく、Android StudioのショートカットやCVSとの連携の仕方、ビルドのタイミングなど、細かい部分の発見が多いです。</p> <p>そこで、私自身がコードを書く時に何を考え、どういう風子に作っているかをライブコーディングでお見せしようと思います。初心者向けで考えていますが、ある程度慣れた方が見ても何個かは新しい発見があると思います。</p> <p>以下の内容に触れる予定です。</p> <ul style="list-style-type: none"> - Android Studioでプロジェクトを作る - Newコマンドを使いこなす - プロジェクトテンプレートを作る - ファイルテンプレートを作る - リネーム、リファクタリングなどの操作系ショートカット - ファイル・エディタ移動、検索などの移動系ショートカット - レアウトエディタの使い方 - ビルド時間を無駄にしない開発方法 	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	konifar

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 14:24:07	How to search and improve performance	<p>▼対象者 初心者～中級者</p> <p>▼概要 Androidアプリ開発で遅くて通れないパフォーマンス改善ですが、細かいtipsなどの話がさがれがちです。どこから手をつけていいのかわからず調べていくのかわからずあまり多く語られていません。</p> <p>本セッションでは実際のオープンソースのアプリ(Android N Easter Egg Neko)を題材に、パフォーマンスが悪化した際にどのように調査するか (Android Studioなどのツール使い方)から「実際のコード改善」までの一連の流れをお話します。皆さんが自分のプロジェクトで実際にパフォーマンス改善できるようになる知見を共有するのがゴールです。</p> <p>また、初心者でも手をつけやすいよう、普段から開発者が慣れ親しんでいるAndroid Studioやadbのツールを取り上げたいと思います。</p> <p>▼取り上げる予定のツール ・adb systrace ・Android Device Monitor</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	Fukui Atsuko
2016/11/01 23:59:00	Android Wear 2.0 のアプリ開発	<p>Google I/O 2016にてAndroid Wear 2.0が発表されました。これはAndroid Wearが発表されてから初の大幅アップデートです。スタンドアロンで動作するようになり作れるアプリの幅が広がり、Android Wearのマテリアルデザインが公開されるなど、Android Wearはさらに面白いものになってきています。</p> <p>本セッションでは、Android Wear初期から個人でAndroid Wearアプリを開発してリリースした知見、Android Wear 2.0 のアプリの作り方をお話します。</p> <p>- Support Wearable LibraryのView解説 - Ambientモード、画面の形状対応 - Android Wearアプリのデザインについて - Android Wearアプリ開発Tips など</p>	ハードウェア (Hardware)		30分 / 30 minutes	teshi04
2016/10/31 15:20:36	テスト0から目指すクラッシュフリー率99%	<p>「なんで、テストカバレッジ0の私がクラッシュフリー率99%台に!?」 様々な現場でAndroid開発をしていると、テストコード皆無のプロジェクトに遭遇することもあるだろう。テストコードがないと大胆な変更をリリースするのに勇気がいる。勇気があるからリリースが怖くなる。リリースが怖いから変更が溜まっていく...の悪循環だ。テストコードはあった方がいい。そんなことは分かっている。それでもないものはない。</p> <p>本セッションではスピーカーの経験から、次のような内容を扱う ・テストコードゼロの現場では何から始めるか ・クラッシュフリー率を上げるには何が効果的か ・テストを書かない現場にどうやって文化を根付かせるか</p> <p>対象は特にAndroid開発を始めたばかりの初心者や、テストコードがなくて苦しんでいる現場のプログラマーを想定している。</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	@fushiroyama
2016/10/31 15:26:22	実践アニメーション	<p>## 対象者 - 初級～中級者 - アニメーションに苦手意識がある人</p> <p>## 概要 Android Viewに対するアニメーションを実装する方法は様々ですが、基本は諸々のサイトで解説されています。しかし、その多くは縮小や移動またはそれらの組み合わせといったシンプルな実装の紹介に留められており、複雑なアニメーションを実現する際の知見はあまり見られません。複数の要素が相対的に異なる動作をするViewなど、実践的なアニメーションをどのように実現するのか、..... 本セッションではアニメーション実装へのアプローチや手法などを具体的に紹介することで、今まで何となくアニメーションにネガティブなイメージを持っていた方々に苦手意識をなくしてほしいと思っています。</p>	UI-デザイン (UI and Design)		30分 / 30 minutes	Naoya Yunoue
2016/10/31 15:59:55	Viewを動的に変化させるアプローチ	<p>## 対象者 アプリのViewにユーザーの操作に追従する動きを入れたい人</p> <p>## 概要 HoloやMaterialデザインが発表されて以降、Viewを動的に変化させることが多くなりました。Viewを動的に変化させるといっても単にアニメーションで動かすだけでなく、最近ではスクロールのようなユーザーの操作に追従して変化させるケースも増えています。 単一のViewを変化させるだけならばViewクラスのsetTranslationXメソッドやsetScaleXメソッドなどを使用すれば可能です。しかし、レイアウト全体を変化させたい場合、Javaのコードだけで行おうとするとなかなかの量の可読性の悪いコードを書かなくてはなりません。更に、その際にAndroidフレームワークの邪魔になる書き方をしてしまうとレイアウトが崩壊する危険性があるため、そうならないように注意して実装する必要があります。 本発表ではそれらの課題について、どのようにすればユーザーの操作に追従したViewの変化を実現しつつ、可読性を保ったコードが実装できるかについてのアプローチをお話いたします。</p> <p>## 発表の構成 - レイアウトXMLの考え方 - Javaからコードを割り込ませる方法 - DataBindingを使う理由</p> <p>## 関連するキーワード - レイアウトxml - DataBinding</p>	UI-デザイン (UI and Design)		30分 / 30 minutes	Takao Sumitomo
2016/10/31 16:20:45	大規模アプリのリノベーション	<p>「はてなブックマーク」の Android 版アプリは、2011年にリリースして以降、大きな変更を行うことなく新機能のリリースや、不具合の修正を続けてきました。しかし初回リリースから5年が経過した今、アプリの規模は非常に大きくなり、対応にコードの複雑性も増し、コード改善の必要性を感じるようになりました。</p> <p>しかし長年積み上げられてきたコードの複雑性を解消するためには、小手先のコード改善ではもはや大きな効果はありません。根本からの見直しが必要です。アプリのアーキテクチャを見直し、旧コードをすべて新たにし、モダンにして行かなければなりません。これはもはやリファクタリングではなく、リノベーションと呼べる非常に大きなものです。</p> <p>リノベーションにあたり、まず初めに取り組んだのが「画面の確認」でした。どの画面がどこから来ているのか。この画面にはどのような機能があるのか。この画面のデザインの意図は何か。ある画面とある画面の共通の View はなにか。叩いている API は何なのか。そのモデルクラスは何か。すべてを一つずつ再確認し、機能を整理し、新たな名前をつけ、それらをドキュメントに落とし込んでいきました。</p> <p>次に取り組んだのは新たな設計の考案でした。まずはライブラリの選定を進めました。近年の Android 開発では有用なライブラリが数多く存在します。例えば Rxjava, Realm, DataBinding があります。これらは開発に大きなメリットをもたらしますが、これらのライブラリには設計を大幅に左右する特性があります。Rxjava を採用すればリアクティブプログラミングがアプリ中に適用されたものになりますし、Realm を採用すれば、Realm で引いたデータを常に監視する設計が考えられます。また DataBinding を利用する場合、自然と MVVM にたどり着くこととなります。</p> <p>つまりライブラリを選ぶということは、同時に設計の形も選ばないといけないのです。そのためにアーキテクチャの選定もほぼ同時期に行うことになりました。世の中には様々なアーキテクチャがありますが、どれを選択すべきなのかは非常に難しいものでした。アーキテクチャをそのまま当てはめようとしても、どうしても Android フレームワークや、アプリの機能、ライブラリの特性にそのままでは合致しないケースがあり、ただアーキテクチャをそのまま選択すればよいというものではありませんでした。</p> <p>設計の難しさは、これらの組み合わせにあります。Android のフレームワーク 様々なライブラリ、そして提案されているアーキテクチャ、これらを組み合わせ、ベストな設計を作り出さなければなりません。勿論その設計は変更しやすく、メンテナンスがし易いものでなければなりません。加えて、我々のアプリでは「5年先も機能を拡張し続けることのできるアプリ」を目指すという目標があります。</p> <p>今回のリノベーションでは、アーキテクチャをそのまま利用するのではなく「アーキテクチャの良いところを可能限り利用する」とも、どうしても実装にそぐわないアーキテクチャの定義は、少しアレンジする」という手法で、設計に落とし込むことになりました。そしてこの設計もすべてドキュメント化し、今後の開発方針がふれられないように徹底しました。</p> <p>本セッションでは、大きなアプリケーションを作り変えるために行った手法について話します。また、Android のフレームワークと適合したアーキテクチャを模索した結果、導き出した設計についても話したいと思います。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	北村 涼

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 17:18:35	4年経くアプリにおけるチーム開発	<p>■対象者 初心者～中級者。 最近Androidアプリ開発を初めた人やチームで開発を初めた人。</p> <p>■内容 フリマアプリ「フリル」は今年で4周年を迎えました。その中でも何度か大きなリニューアルを実施してきました。</p> <p>コードベースも必然と大きくなり、複数人で大規模アプリのメンテナンスをし続けながら、機能追加などをしてきました。</p> <p>このセッションでは4年間の中で、チーム開発がどのように変化してきたのか、どういったことをやってきたのかをお話します。</p> <p>・開発フロー ・開発環境 ・コードレビュー ・フリクエスト ・CIの連携 ・社内ユーザーインタビュー ・QAのやり方 ・リリース ・数値計測 ・リニューアル ・レビューマネジメント</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	cutmail
2016/10/31 18:06:42	Kotlin + RxJava + Dagger2 + Orma + Retrofit で作るAndroidアプリ	<p>対象: 初心者～中級者</p> <p>- Kotlin(Language) どうかの? - RxJava(ReactiveProgrammingFramework) どうかの? - Dagger2(DependencyInjection) どうかの? - Orma(ORMapper) どうかの? - Retrofit(HttpClient) どうかの?</p> <p>そんな疑問にお答えします。(個人的にはもはやこの5つのソフトウェアはAndroidアプリケーション開発において無くてはならない存在です。これらを利用したAndroidアプリケーションの作り方を紹介したいと思います。</p> <p>目次(予定) - 各ソフトウェアの紹介 - Kotlin? - RxJava? - Dagger2? - Orma? - Retrofit? - DroidKaigi2016公式アプリのコントリビューター 一覧画面を作ろう - RetrofitをつかってApiClientを実装しよう - OrmaをつかってDaoを実装しよう - Daggerを使って依存性を注入しよう - ApiClientとDaoを束ねるRepositoryをつくろう - UseCaseをつくろう - Presenterをつくろう</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	@lvla0805
2016/10/31 18:12:24	APIファーストで始めるアプリ開発	<p>対象者: アプリ開発者でサーバーサイドWebAPIと仕様調整をしなければならないエンジニア(中級者)</p> <p>内容: あるサービスをモバイルアプリで提供するにあたり、サーバーサイドWebAPIの存在が不可欠です。しかしながら、既存稼働中のサーバーアプリや新規開発では、アプリ開発時点ではWebAPIが存在しない事が多々あります。モバイルアプリとしてWebAPIの仕様調整をサーバー担当者を行うにあたり、OpenAPI仕様のSwaggerを活用したアプリ開発について紹介します。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Katsumi Onishi
2016/10/31 18:16:45	Exploring new Android layouts	<p>Creating optimized layouts is a stubborn task, Android developers need to spend considerable amount of time on it because there are some common pitfalls with existing layouts (LinearLayout, RelativeLayout, etc) and there are lots of screen sizes they need to care considering the device factors (screen size, density, orientation, multi window mode from Nougat).</p> <p>In this talk I'm going to cover the two new layouts that were introduced from Google in 2016, ConstraintLayout and FlexboxLayout. The topics will include Android views in general, what are the common pit falls of existing layouts, basics of the new layouts and what common layout problems they try to solve.</p> <p>Disclaimer: the presenter is the author of the FlexboxLayout, but I'll try to explain the unbiased pros and cons regarding the existing and newly introduced layouts from technical point of view.</p>	UI・デザイン (UI and Design)		50分 / 50 minutes	thagikura
2016/10/31 18:20:06	InputMethodServiceとはじめ	<p>対象: Androidでキーボードを作りたい方 InputMethodServiceの内部を知りたい方</p> <p>概要: スマートフォンを使う上でIMEを全く触らないという日はおそらくないと思われます。しかし、IMEを開発するとすると、Android Developers以上に情報がまとまったものはなかなか見つかりません。そこで、本セッションでは、AndroidのInputMethodServiceのライフサイクルやAPIIに関して共有できればと思います。Android 7.1で導入されたImage Keyboard Supportもカバーする予定です。</p>	プラットフォーム (Platform)		30分 / 30 minutes	yoshiki obata
2016/10/31 18:30:04	What is tested by pre-launch (security) reports?	<p>アルファ版、ベータ版としてapkファイルをGoogle Playにアップロードする際に、pre-launch reports(日本語訳はリリース前レポート、オプションが必要)によりアプリに含まれる問題点が表示されるようになりました。表示される内容の一つに、「セキュリティ」があり、アプリ中の脆弱性を表示してくれます。しかし、どのような脆弱性が検出されるのかについては明かされていません。</p> <p>本セッションでは、リリース前レポートがどのような脆弱性を検出する傾向にあるのかをお教えいたします。また、その結果を踏まえたうえで、リリース前レポートとどのように付き合うべきか考察します。</p> <p>言及する脆弱性は、日本スマートフォンセキュリティ協会が発行している「Androidアプリのセキュア設計・セキュアコーディングガイド」で言及されている内容を元に調査したものととなります。なお、調査を行った時点での内容となります。</p> <p>■対象者 -Android開発初心者～中級者 -Androidアプリの脆弱性対策に興味のある方</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Akihiro Shiota

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 18:38:32	How to implement material design animation	<p>■対象 中級者向け。 アニメーションの実装をある程度行ったことがあり、マテリアルデザインのアニメーションの方法は知らない人向け。</p> <p>■内容 Googleから発表されたデザインガイドライン マテリアルデザイン(Material Design)は話題になり、多くのアプリに取り入れられています。しかしアニメーションの実装方法まで詳しく取り入れられているアプリはとて少ないのではないかと考えています。 またマテリアルデザインガイドラインでは、デザインのガイドラインであるため、Android上でどう実装するかまでは触れられていません。</p> <p>Googleはマテリアルデザインの例としてPlaid, Topeka, OurStreets, Google I/O 2016などのサンプルアプリを出しており、どのように実装するかを調べることができます。 私はこのマテリアルデザインのアニメーションの実装方法を調べ上げ、またSupport Libraryを用いた下位互換性や対応方法も調べます。 実践的にマテリアルデザインのアニメーションを実装できるようになっていただきたいと思います。</p> <p>■大まかなセッションの内容(案) マテリアルデザインガイドラインの"マテリアルモーション"をどのように実装していくのか、そのAPI Level, 低いAPIでの回避案などを解説していきます。</p> <p>* "マテリアルの動き方"を実装するには * "優れた遷移の特徴"を実装するには * アニメーションの"継続時間"の実装 * "モーションアイコン"の実装方法</p> <p>などなど</p>	UI-デザイン (UI and Design)		30分 / 30 minutes	takahiro
2016/11/01 22:50:14	Android Studio爆速テクニック	<p>■対象 初心者向け。 Android Studioを使ったことはある人。</p> <p>■内容 みなさんAndroidアプリの開発をするときはAndroid Studioを利用して開発をしていると思います。そこでどんなふうに開発するでしょうか？ショートカットは何を使っているでしょうか？ショートカットやプラグインはどのようなものを利用していますか？ 様々なテクニックを知っていると速く効率的にコーディングができますが、効率的なテクニックを知る機会が少なく、できていない方が中々多いと感じています。</p> <p>ライブコーディングを通して実際に体感しながらおすすめのテクニックを紹介し、聞いた方の皆さまがライブコーディングを通して基本的なショートカットからはじめ、爆速で開発できる状態を目指したいと思います。</p> <p>■大まかなセッションの内容(案) 基本編 ショートカット</p> <p>* Find Actions * QuickFix * Basic completion & Smart type completion * ショートカットを使ったVCSの使い方</p> <p>などなど</p> <p>応用編</p> <p>* インスペクション(Lint, Find bugs, CheckStyle) * Android StudioのVCSコミットダイアログを利用したインスペクションの運用方法。 * .ideaファイルを改竄する * 生産性を上げるプラグイン - Key promoter - Android Postfix Completion</p> <p>などなど</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	takahiro
2016/10/31 18:52:09	実践Camera2	<p>Lollipop より導入されたCamera2 APIですが、公開されてから2年以上たった今でもあまり情報が見当たらない状況です。またハードウェアに近いAPIであるために機種依存の問題が多く、さらにこのAPIを使った実装を困難にしています。</p> <p>このセッションではそのCamera2 APIを実際の製品に導入した経験から、</p> <ul style="list-style-type: none"> - Camera2 APIの概要 - Camera2 APIを使ってできること - RxJavaをつかった非同期APIの実装 - Camera2 APIにおける機種依存の問題 - AWS Device Farm / Firebase Test Labを使って機種依存の問題を発見する手法 - 機種依存の問題を解決する手法 - Camera2実装のベストプラクティス <p>について紹介したいと思います。</p>	ハードウェア (Hardware)		50分 / 50 minutes	山本純平 (@boohbah)
2016/11/01 22:52:06	Androidフレームワークコードリーディングに必要な知識	<p>■対象 中級者向け Androidのコードを読みたいがよく挫折する人向け。</p> <p>■内容 AndroidはAOSP(Android Open Source Project)としてソースコードが公開されています。AOSPのソースコードを読むことでアプリ開発に役立つたくさんの知識を得ることができます。しかしAndroidのフレームワークを読んでいく上で障壁となる点が多くあり、コードリーディングをしていく上でいくつかコツがあります。</p> <p>例えば以下のような点です。</p> <ul style="list-style-type: none"> * Binderによるプロセス間通信の向こう側のクラスを見ていく方法 * 抽象化されていて実装が見れないWindowの裏側 * Support LibraryのAppCompat系のViewの置き換え <p>またどのようなツールを使ってコードリーディングをしていくのかも最初は難しい課題となります。そこで必要となる知識を体系的にまとめて紹介します。</p> <p>■大まかなセッション内容(案) 読む上で利用するツール ・AOSP(gitiles) ・AOSP(Git) ・Android Studio ・Android SDK Search など</p> <p>ソースコードを読む上で理解しておく便利な知識 ・ActivityとWindowとViewの関係 ・Support Libraryが何をしているのか？ ・Binderによるプロセス間通信の仕組み、読み方 など</p> <p>ライブコーディング 知識をまとめ、他の人に知識を共有する方法</p>	プラットフォーム (Platform)		50分 / 50 minutes	takahiro

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 19:10:06	Support Libraryに含まれるイマドキのUIコンポーネントを使いこなす	<p># 対象者</p> <ul style="list-style-type: none"> - Androidアプリ開発初心者で、Support LibraryにどのようなUIコンポーネントが用意されているか知りたい人 - 最近Support Libraryに追加されたUIコンポーネントについて知りたい人 - Material Designを実現するにはどのようなUIコンポーネントを使えばよいのか知りたい人 <p># 概要</p> <p>今やAndroidアプリ開発になくてはならないSupport Library、その中にはAndroidに最適なUIを実現するための様々な便利なUIコンポーネントが収録されています。特にバージョン22以降では、Googleの提唱するMaterial Designのデザインパターンを実現するためのUIコンポーネントが数多く追加されています。このセッションではSupport Libraryに含まれるUIコンポーネントの中から、最近(バージョン23-24以降を予定)追加されたものの使い方や使う際のtipsを紹介します。内容は可能な限り発表時の最新バージョンに準拠する予定です。</p> <p># 構成案</p> <ul style="list-style-type: none"> - Support Libraryについて軽く紹介 - Support LibraryとMaterial Designの関係性について - Support Libraryに含まれるUIコンポーネントを使う - UIコンポーネントを使う上でのtips 	UI-デザイン (UI and Design)		30分 / 30 minutes	Takuji Nishibayashi
2016/11/01 23:54:23	ゼロから始めるRxJava 2.0	<p>#対象者</p> <p>Android中級者 RxJava未経験者 RxJava 1.0初心者</p> <p>#概要</p> <p>AndroidでRxJavaを利用することは今ではかなりメジャーになってきました。RxJava導入の難点としてよく挙げられるのが学習コストの高さです。また、RxJava2.0ではObservableから排出される値数やBackpressureを意識したプログラミングが必要になるなど、さらに少し敷居が高くなった印象があります。そこで本セッションでObservableとは何かなど、かなり基礎的な部分から説明を行い、RxJava触れたことがない初心者が、RxJava 2.0を軽く理解するところまでもっていきたいと思っています。</p> <p>#大まかなセッションの内容(案)</p> <p>なぜRxJava? RxJavaを構成する要素(Observableって何? など) 組み合わせるとどんなことができる? リアクティブプログラミングがあるコードとないコードのパフォーマンス RxJava2.0でどう変わった? (Flowableって何? など) Backpressureって? Hot? Cold? などなど</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	takahiro
2016/10/31 19:53:39	Taking care of your UI tests	Most Android UI testing talks focus on how Espresso works and how to start writing simple tests. This is fine for a simple app but maintaining and expanding those tests gets harder as the app grows. This talk will focus on writing complex tests scenarios while making sure we keep them maintainable, extendable and readable. I'll use a released app we developed as an example. This will involve multi-screen scenarios, handling asynchronous operations, providing a fake environment as well as some common and less common Espresso gotchas.	メンテナンス (Quality and Sustainability)		50分 / 50 minutes	Florian Mierzejewski
2016/10/31 19:54:10	DaydreamアプリをUE4で開発する方法 & JAGOVISOR/jagoButtonの紹介	<p>DaydreamアプリをUE4で開発する方法&JAGOVISOR/jagoButtonの紹介</p> <p>内容(予定): ※ Googleのプログラクリリースによって変更される可能性があります 2017年は、VR市場二年目で多くの展開が予想されます。VRデバイスの第二世代が続々とデビューし、AndroidではDaydream対応端末が各社から出揃います。VR市場の現状のトレンドをまとめるとともに、UE4を使ったDaydreamアプリの開発入門、VRアプリ作成時に配慮すべきポイントを解説します。また、後半では、日本Androidの会 VR部・金沢支部で開発を行っているVRゴーグル"JAGOVISOR"の紹介と、その機能の1つである超低コストで実現された有線接続物理3(6)ボタンのインターフェイス"jagoButton"対応アプリケーションの開発方法を紹介します。</p> <ul style="list-style-type: none"> ・VR市場トレンド ・Daydreamの概要 ・VRアプリケーション開発における注意事項 ・Unreal Engine 4を使ったDaydreamアプリケーションの開発方法の解説 ・日本Androidの会 VR部・金沢支部で開発を行っているJAGOVISOR/jagoButtonの紹介 ・JAGOVISOR:組み立て簡単、IPD調整が可能なVRゴーグル ・jagoButton:マイク端子に接続する、有線接続でペアリング不要、SDK追加不要の物理ボタン ・jagoButton対応アプリケーションの開発方法の解説 (Unreal Engine 4を使って) <p>対象:</p> <p>非プログラマ(デザイン・企画担当者)、これからVRアプリの開発をはじめの方</p>	その他 (Other)	VR, Daydream	50分 / 50 minutes	kinneko 日本Androidの会 VR部・金沢支部
2016/10/31 19:58:46	人工知能によるAndroidアプリの自動テストの試み	<p>#対象者</p> <ul style="list-style-type: none"> ・Android アプリ開発の経験のある方 ・人工知能に興味のある方 <p>#概要</p> <p>モバイルファーストの到来により、これまでの「PC中心のシステム開発⇒利便性向上へ向けた追加のモバイル開発」から、モバイル先行の開発、及びモバイル専用のシステム・アプリの開発が中心となってきており、グローバルに見ても、モバイル市場の成長は目を見張るものがあります。スマートフォンアプリの開発現場においては、その開発スピードが求められているにも関わらず、増え続ける機種数×OS数により、テスト工程、とりわけAndroid端末における実機検証が大きな課題となっています。このような課題を解決するため、人工知能による実機検証を自動化する試みを行っています。アプリをアップロードするだけで実機検証を開始し、クラッシュ情報や、操作履歴を学習しながら、精度を高めていく完全自動化を目指した実機検証サービスの構想、検証、効果、展望などを紹介させて頂きます。</p> <p>#目次案</p> <ul style="list-style-type: none"> ・モバイルアプリ開発における課題 ・課題の解決策 ・人工知能による自動テストの可能性検討 ・人工知能を実現するためのアプローチ ・人工知能の実用化の検証 ・学習し成長し続ける人工知能の展望 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	ShuipingLi
2016/10/31 20:53:10	Don't blame the user	As developers, we tend to focus on the quality of our code, working with the cleanest of architectures and best coding practices. As designers, we follow the Material Guidelines to ensure we are consistent with the platform, manipulating motion in our designs to get the message across. Does that really give any value to our users though? Are we really working towards making the life of our users easier? In this talk, Alex will discuss how to improve the overall app rating without writing a single line of code or do any design. Instead we will see how to have a better understanding of our user-base by analyzing user reviews, doing user interviews and other methods and how to rise up the true issues the users are trying to solve by using your app. Can make talk 30-50 mins, flexible.	その他 (Other)	Users (User Research, UX, Con	50分 / 50 minutes	Alex Styl
2016/10/31 21:12:22	Reverse engineering is not just for hackers!	<p>We spend a lot of time putting apps together, but when was the last time you pulled one apart? How wonderful is it that Android is open-source, so we can simply look at the code when we need to? What if it were just as easy to look at the source code and behaviour of any other app?</p> <p>If we can streamline the process of looking inside a compiled application then we're more likely to employ it to answer questions and teach us valuable lessons we can apply to our work. We may learn from others and also make our own apps more secure. We can pinpoint bugs that come from closed-source libraries such as those for ad and tracking networks, and work around those bugs, get them fixed faster, or even patch them if need be!</p> <p>This talk will present simple real-world examples for maximum practical benefit using some of the ever improving set of reverse engineering tools for Android.</p> <p>You don't need to have any experience reverse engineering anything before, but hopefully even if you do you'll pick up a few useful tips. This talk aims to make every developer more familiar with the reverse engineering tools available for Android, and how and why they should apply them. There's an incredible amount that can be learned from taking things apart!</p>	その他 (Other)	Reverse Engineering	30分 / 30 minutes	Jon Reeve

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/10/31 22:03:25	FluxによるリアクティブなAndroidアプリケーション開発	<p>クライアントサイドのアプリケーション開発が複雑になる要因として下記が挙げられます。</p> <ul style="list-style-type: none"> - ユーザイベントの存在 - WEB-API通信など非同期的な処理 - アプリケーションが状態を持つ - 表示に関わるコードの増大 <p>本セッションでは複雑さを解決するアプローチのひとつとしてFacebookが提唱したFluxアーキテクチャに着目し、Androidアプリケーションの開発に適用してみても得られた知見について発表します。</p> <p>■Fluxについて(Qiita) AndroidでRxJavaを利用してFluxしてみた - http://qiita.com/hkusu/items/64a9435e1613e4c20ba7</p> <p>■Fluxの参考実装 https://github.com/hkusu/android-rx-flux-example (採択されれば当日までにブラッシュアップします。)</p> <p>■話すこと案</p> <ul style="list-style-type: none"> - フロントエンド(JavaScript)界隈の話(MVC2/MVVM/Flux、React) - Fluxアーキテクチャ概要 - 解決したい問題 - Androidでの実装の概要 - 実際にFluxを採用して良かったこと/悪かったこと - テストについて - 気づき、Tips - 参考実装の紹介 <p>主にコードベースで説明します。ライブラリとしては RxJava、Data Binding、Dagger2(DI)というよりインスタンスの管理/依存を利用することを前提にしているので、これらのライブラリを利用する目的および解決する問題についても触れます。聴講された方が後から評価/導入できるよう、当日までに参考実装を用意してGitHubに置きます。</p> <p>このアプローチは銀の弾丸ではなく、また本発表は他のアーキテクチャとの優劣をつけるものではありません。例えAndroid界隈的に悪い評価、失敗例となったとしても、今後でてくる複雑さに対応するアプローチの礎、議論の種になれば幸いです。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Hiroyuki Kusu (@hkusu_)
2016/10/31 22:32:31	Can You Read Your Tests? Clean and Useful Android Testing, with JUnit and Spock!	<p>Can make talk 30-50 mins, flexible.</p> <p>Have unit tests? How readable are they? How much time do you spend maintaining them? Brittle and time-consuming tests lead to fear, uncertainty, and distrust, and that leads to and eventual disillusionment and abandonment.</p> <p>This talk will go over tools and techniques for writing tests that are a pleasure to read, easy to maintain, give you maximum return on the time you invest into them, and will prove their worth time and time again.</p> <p>We'll take a look at some example tests that are exhibiting some common issues, such as:</p> <ul style="list-style-type: none"> - readability - brittleness: breaking every time the implementation is changed - non-obvious cause + solution upon test failure - flakiness / unreliability - slow execution <p>We'll improve these tests in plain JUnit, move on to Hamcrest and AssertJ, and then show how Spock can take us even further to tests that are beautiful, easily maintainable, and incredibly useful.</p> <p>Stop wasting time on bad tests!</p> <p>This talk aims to level-up every Android developer's testing know-how and arm them with the tools required to write effective and maintainable tests.</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	Jon Reeve
2016/10/31 23:38:52	ブラウザで気軽にアプリの動作確認をしよう!	<p>■対象者 ブラウザでも気軽にAndroidアプリの動作確認などを行いたい人 初級~中級向け</p> <p>■概要 ブラウザでAndroidアプリの動作確認をする方法をご紹介します。</p> <p>具体的には Appetize を使ったブラウザでのAndroidアプリの動作確認を行うまでをご紹介します。こういった Appetize のようなサービスを使った場合、ログインなどがあるアプリの場合だと毎回ブラウザ内のアプリでログインが必要になると思うのですが、それを回避して如何に簡単にブラウザでAndroidアプリの動作確認を行うかを突コードを交えながらお話しします。</p> <p>また、Appetize の他にも、Chromeの拡張機能を使って動作する ARChon などについてもご紹介します。</p> <p>内容案:</p> <ul style="list-style-type: none"> * Appetize の簡単な使い方 * Appetize を使ったときのセッションの動機方法 * ARChon を使ったChromeでのAndroidアプリの動作方法 <p>■キーワード Appetize, レビュー, プルリクエスト, 動作確認, ブラウザ</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	henteko
2016/10/31 23:59:40	Chrome Custom Tabsをさらに使いこなそう	<p>■対象者 初級者~中級者</p> <p>■概要 Chrome Custom Tabs(以下CCT)がSupport Libraryの23.0.0から追加されました。CCTの存在を知っている人が、なかなか導入できない人もいます。出た当初に比べて、現在はできることも少しずつ増えてきています。CCTでできることできないことを把握して、導入できるか再検討してみましょう。既に導入している人も、新しい知見を持ち帰っていただきたいと思います。</p> <p>■目次案:</p> <ul style="list-style-type: none"> * CCTとは * 導入方法 * 外見を変える * アニメーションをつける * Referrerをつける * Headerをつける * BottomBarをつける * RemoteViewをつける * Prefetchで素早く読み込む * イベントを受け取る * Sessionをつなぐ <p>■補足: あまりコードには触れず、できることできないことにフォーカス置こうと思っています。事前にサンプルコードは公開し、それに沿う形にします。余裕があれば(調べきれれば)、Chrome側から見たアプリにも触れたいと思っています。</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	sakebook
2016/10/31 23:59:49	Android Bikeを作ろう	<p>Android Autoはちょっと手が届かない...けどBike(自転車)なら!</p> <p>「最近購入した自転車にBluetoothを喋るので何かやってみよう」Androidならそのぐらい気軽にIoTできるよ、ちょっと頑張れば既にある製品より圧倒的に使いやすい機材を自分で作れるよ、という話を、ある程度Android開発に慣れてきた開発者向けにお話しします。身近なBluetooth Low Energyデバイスや実はあんな家電もこっそり生えてるAPIに、スマホのユーザーインターフェースを組み合わせることで、手軽に自分専用の道具を作ることができます。このセッションが、普段開発しているスマホやタブレットという枠を超えた外の世界とインターネットを繋いで、世界にカジュアルに新しい価値を提供していくきっかけになると幸いです。</p>	ハードウェア (Hardware)		30分 / 30 minutes	tnj

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 0:39:21	React Nativeはクロスプラットフォームモバイルアプリ開発の夢を見るか	<p>2015年最初にオープンソースプロダクトとして登場したReact Nativeは、現在も活発に開発が続いています。Facebook Messengerだけではなく、プロダクションでの事例も少しずつ出てきました。React Nativeを使うと、何が嬉しいのでしょうか。実際に業務でAndroid、iOS両対応のアプリを開発・リリースした経験から、Android Javaでの開発と比べて楽になる点や難しくなる点をお話します。</p> <p><対象></p> <ul style="list-style-type: none"> * 最近のJavaScriptはどんな空気で開発をしているのか見てみたい方 * 心のどこかでまだReact Nativeを銀の弾丸だと思っている方 * AndroidとiOSのコードはどこまで共通化できるのを知りたい方 * Webフロントエンドのコードとどこまで共通化できるのを知りたい方 <p><話したいこと></p> <ul style="list-style-type: none"> * NodeJS, NPM文化圏とES2015+CSS3がもたらすパラダイム <ul style="list-style-type: none"> - Promise - fetch - module - 様々なJSライブラリ - Flexbox * FlowTypeでJSなのに静的型付けできた話 * Hierarchy Viewerで分かった、Virtual DOMの先にあるAndroidネイティブの姿 * Webフロントエンドエンジニアたちとライブラリを共有できて最高だった話 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	中川幸哉 (@Nkzn)
2016/11/01 22:12:21	変更に強いEspressoテストコードを効率良く書く	<p>■対象者</p> <ul style="list-style-type: none"> * Espressoに興味は有つつも、取り組み時間がなくて実行に移せていない方 * 効率的に、メンテナンス性の高いEspressoテストコードを書きたい方 <p>■概要</p> <p>AndroidのUIテストツールであるEspressoは、Android Studio 2.2から導入されたEspresso Test Recorderによって、テストを書く数居が大きくなりました。</p> <p>しかしながら、単純にEspresso Test Recorderに頼ってテストを量産すると、重複したコードが多数出現してしまい、画面仕様が変更された時に手が付けられなくなってしまいます。</p> <p>このようなメンテナンス性の問題を解決するためには、量産したテストコードを適切に共通化する必要がありますが、量産してしまったコードを後から共通化するのも骨が折れる作業です。</p> <p>このセッションでは、業務でテストの書き方を支援している自身の経験にもとづいて、効率的にテストコードを生成できるEspresso Test Recorderの利点を活かしつつも、メンテナンス性を損わないテストコードを書く方法について、ライブコーディングを交えながら解説します。</p> <p>主に、以下のような内容を考えています。</p> <ul style="list-style-type: none"> - Espressoテストコードの基本 - Espresso Test Recorderの限界を理解して、うまくテスト自動化対象を選定する方法 - Espresso Test Recorderで何を記録すべきか - 記録したテストコードを元に、メンテナンス性を損わずにテストを量産する方法 (Android Studioのリファクタ機能を活用します) <p>個々の要素は基本的なことではありますが、「どのように組み合わせるとまくいのか」という知見を皆さんと共有することで「テストを書いてみよう」と思って下さる方が増えるような内容にしたいと考えています。</p>	メンテナンス (Quality and Sustainability)		50分 / 50 minutes	外山純生 (sumio_tym)
2016/11/01 2:29:51	Revving Up Rendering: Android Hardware Acceleration	<p>Hardware acceleration, where rendering occurs on the GPU, has been available on Android for some time (since API 11). While hardware acceleration may often work silently without your knowledge to bring better rendering to your views, you may find situations where you either want to disable it or take more control of it to improve the performance of complex drawings and animations.</p> <p>In this talk we will review how hardware acceleration works, starting with a review and comparison of the software-based drawing model and hardware accelerated drawing model. We will then explore how hardware acceleration can affect your custom views and drawing, show how to enable/disable hardware acceleration, and work through an example of using view layers for increased performance.</p>	プラットフォーム (Platform)		30分 / 30 minutes	Huyen Tue Dao
2016/11/01 2:40:51	Augmenting Android with Firebase	<p>Firebase has evolved from an amazing realtime database to include a full suite of BaaS features. This talk will help you learn where to start and which features might be right for your Android application. We will learn about topics including easy SSO authentication with Google, Twitter, Facebook, etc. We will also cover remote configuration for A/B testing and feature flags, and finally we'll explore the value of the realtime database and file storage features.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	Joshua Jamison
2016/11/01 2:55:11	Don't reset --hard: Strategies for Tackling Large Refactors	<p>How many times have you started an ambitious refactor only to get lost and end up doing a git reset --hard? Android libraries are updated constantly, sometimes with breaking changes, and it can be tough to keep up. Maybe you want to try several new technologies at once as part of your refactor. This talk will teach you some techniques for refactoring your code in a way that makes you not get so overwhelmed that you have to start over.</p> <p>This talk will focus on a refactoring strategy made popular by Sandi Metz, a prominent figure in the Ruby community. Even though writing code for Android is quite different from writing Ruby, the same techniques can be applied towards tackling a large refactor in any language or framework. I will discuss concepts such as "shameless green" (keeping a green test suite or other sanity check as you refactor), and detail how layering on your refactor in stages can be extremely effective in keeping you on track and moving you closer to your goal without breaking everything. It can be hard to keep up with new Android technologies, but a good refactor can be a great introduction to a new library or concept. This talk will contain a practical example of a refactor that introduces several new technologies to an existing Android app.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Siena Aguayo
2016/11/01 3:06:23	Realtime Data Deep Dive with Firebase	<p>One of Firebase's most powerful services is an amazing realtime database for synchronizing your data across multiple devices and users. The database structure and usage is quite different from a standard SQLite DB which can make it intimidating to set up. In this talk we'll cover how to structure your data using Firebase's NoSQL backend. We'll also learn how to monitor and modify the data. Finally we'll explore techniques of securing the data using rules.</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Joshua Jamison
2016/11/01 3:26:54	Material Design with Remixer	<p>As many developers are aware, Google has standardized design on Android with their Material Design guidelines. While many apps have embraced the new standard, tweaking and improving UI can still be a constant struggle for both developers and designers. Google recently announced their Remixer SDK that greatly improves that workflow for changing and prototyping UI. In this talk we will explore the capabilities of the Remixer SDK, including how to use it to dynamically change design values such as color, animations, and view sizes. We'll also explore how you can change the design of your app without submitting a new app version. This talk will be focused on live demos of changing the design of an example app. The goal is to show attendees how they can improve their process of UI development and save hours by not waiting around for their projects to rebuild for one value change.</p>	UI・デザイン (UI and Design)		50分 / 50 minutes	Caren Chang
2016/11/01 3:41:40	Scoping with Dagger 2	<p>Dagger 2 is a powerful Dependency Injection library that can be implemented in various ways. Scopes are an often over-looked feature that can help increase your control of your data provision and injection. In this talk we'll dig deep into Dagger 2 Scopes with detailed code samples. We'll learn how to set up your Dagger 2 implementation for Scopes, create unique Scopes, and apply them to your injection targets.</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	Joshua Jamison
2016/11/01 9:35:11	逆引き Material デザイン	<p>This talk is for both current Dagger 2 users as well as those who are new to Dagger 2.</p> <p>マテリアル デザインが Google I/O 2014 で発表されてから 3 年近くが過ぎ、今では多くの有名アプリがマテリアル デザインに沿った UI/UX を提供するようになっています。一般の Android ユーザーも当然のようにマテリアル デザインに沿った体験を期待しています。本セッションでは、マテリアル デザインガイドライン (material.google.com) と実際の実装の間に架け橋を築きます。ガイドラインの背景となる考え方を踏まえた上で Android アプリでマテリアル デザインを実装する際の指針を提案し、プラットフォームやサポート ライブラリの各コンポーネントとマテリアル デザイン ガイドラインの関係について議論します。また、Android マテリアル デザイン サポート ライブラリの開発エンジニアの一人として、古いプラットフォームも念頭に入れた実際の実装方法についてお話します。</p> <p>あくまでエンジニア目線のセッションですが、UI/UX を形にしたい上でその考え方を知りたいデザイナーにも役立つようにお話します。より良い UI/UX を実現するための具体的な観点を抑えた上で、それを実際のアプリにするための技術を見ていきます。</p>	UI・デザイン (UI and Design)		50分 / 50 minutes	荒木佑一

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 10:17:59	Kotlinを始めようハンズオン	<p>■概要 新しいプログラミング言語のKotlin(ことりん)を使ったAndroidプログラミングのハンズオンです。長らく(特に)Androidエンジニアの注目を集めていたKotlinですが、今年2月に念願のver1.0のリリースと7月に日本初の書籍の出版があり、今非常に勢いのある言語です。</p> <p>■対象者 ・Android初心者以上 ・Kotlin初心者</p> <p>■ゴール/得られる知識・技術 ・AndroidプロジェクトにKotlinを導入できる ・KotlinをAndroidアプリ開発に活用できる(Kotlinの利点を享受できる) ・Kotlinの弱点を知り、フォローできる</p> <p>■注意 ・ハンズオンということで実際に手を動かしていただきます ・ただし50分という短い時間の中でのハンズオン部分には非常に少ないです ・ハンズオン参加の方は、事前にお知らせする環境をセットアップした状態のノートPCをお持ちください ・ハンズオン不参加でもご聴講いただけます</p> <p>■目次案 1. Kotlin概要(10分) 2. 環境構築～導入(解説5分、ハンズオン5分) 3. アプリ実装(解説10分、ハンズオン15分) 4. 次の一歩(5分)</p> <p>※題材となるアプリは検討中(ごく簡単なツール系を予定)</p> <p>■登壇者について ・日本Kotlinユーザグループ代表 ・講演実績多数: DroidKaigi, JJUG CCC, Hacker Tackle, LLoT, ABCなど ・執筆実績多数: Kotlinスタートブック, Software Design, 日経ソフトウェアなど</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	長澤太郎 @ngsw_taro
2016/11/01 10:30:35	逆引きKotlin 10連発!	<p>■概要 KotlinでAndroidプログラミング..「Javaではああやっていただけ、Kotlinだとどうなるの?」「Kotlinらしいコードを書きたい!」といった声にお応えすべく、目的から機能やコードを紹介する逆引きセッションです。</p> <p>■対象者 ・Android中級者以上 ・Kotlin初心者以上</p> <p>■ゴール/得られる知識・技術 ・KotlinをAndroidアプリ開発に活用できる(Kotlinの利点を享受できる) ・Kotlinの弱点を知り、フォローできる</p> <p>■登壇者について ・日本Kotlinユーザグループ代表 ・講演実績多数: DroidKaigi, JJUG CCC, Hacker Tackle, LLoT, ABCなど ・執筆実績多数: Kotlinスタートブック, Software Design, 日経ソフトウェアなど</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	長澤 太郎 @ngsw_taro
2016/11/01 10:43:36	Javaを使い続ける	<p>■概要 KotlinやGoが台頭してきていますが、やはりAndroidアプリをJavaで書き続けたい!というそんなあなたに贈るJavaライブラリ・ツールの紹介をします。快適なJavaライフを。</p> <p>■対象者 ・Android初心者以上</p> <p>■内容 最近のAndroid Javaの概要や背景を簡単に解説します。そのあとに、Java SE 7以前の言語機能でも快適にプログラミングを行うためのライブラリやツールをひとつずつ紹介していきます。</p> <p>■登壇者について ・日本Javaユーザグループ幹事、日本Kotlinユーザグループ代表 ・講演実績多数: DroidKaigi, JJUG CCC, Hacker Tackle, LLoT, ABCなど ・執筆実績多数: Kotlinスタートブック, Software Design, 日経ソフトウェアなど</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	長澤 太郎 @ngsw_taro
2016/11/01 11:47:03	できる! Android Framework Code Reading	<p>■対象者 Android Frameworkのコードを読みたい人 Androidの内部構造の調べ方を知りたい人</p> <p>■概要 Androidの面白さはオープンソースでコードが公開されているので、Androidがどのように動いているのか?とかをコードを読めば知れることです。Androidの内部構造を知ることで、アプリ開発に役立つこともたくさんあります。しかし、コードが公開されているとはいえ、膨大な量のコードからAndroidは作られています。調べたい機能がどのように動いているのかを調べる際に、いきなり膨大な量のコードを読んでも挫折してしまうことがよくあります。</p> <p>本セッションでは、そもそも「Android Frameworkってなに?」というところから、実際にFrameworkのコードをどのように読んでいくかなどをお話します。具体的な内部構造の話よりも、「どのようにAndroid Frameworkのコードを読んで内部構造を調べるのか」をメインとして扱う予定です。</p> <p>具体的には以下のような項目についてお話す予定です。 * Android Frameworkってなに? * なぜAndroid Frameworkのコードを読むのか * Android Framework Code Readingをする上で便利なツールなどの紹介 * 読みたい目的のコードを探し出す技術 * Android Frameworkのコードを読む時に使える技術 * Android Frameworkでアプリ開発者が手軽に読めそうな部分の紹介 * コードを読んだ結果をうまくまとめる技術</p>	プラットフォーム (Platform)		30分 / 30 minutes	operndoOS
2016/11/01 12:16:27	Doze for the "N"ext	<p>Android Mで紹介されたDoze modeですが、Nでさらに浅いDoze (Light Doze) が追加されプログラマが対応する項目は増えています。弊社アプリではDozeの挙動がアプリの提供できる価値に致命的な影響を及ぼすことから、早くからDozeに対する調査・対応を行っています。本セッションではそれらの過程で培われた知見を共有します。</p> <p>・MとNのDozeに関するおさらい ・Dozeによって受ける制限とそれに対する対応方法の最新状況 ・Tips等</p>	プラットフォーム (Platform)		30分 / 30 minutes	@fushiroyama
2016/11/01 12:18:18	自社開発のタブレット端末でサービスを立ち上げた話	<p>自社開発のタブレット端末でサービスを立ち上げた話</p> <p># 対象 Androidの自社開発のタブレット端末を使ったサービスに興味がある方</p> <p># 概要 ジャストシステムでは「スマイルゼミ」というAndroidタブレット上で学習プラットフォームを提供しています。スマイルゼミは市販のAndroidタブレットを用いず、独自に開発したタブレット端末を利用しています。独自端末を開発し、その上でサービスを展開した知見を共有します。</p> <p># アジェンダ案 - 独自端末を作るとは - プリンストールソフトウェア - 独自端末だからできたこと - 独自端末での苦労話</p>	プラットフォーム (Platform)		30分 / 30 minutes	Yuta Tetsuka

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 12:25:50	Androidリアルタイム通信アプリ作成Tips	<p>ユーザが能動的に画面をロードすることなく更新されていくリアルタイム通信型のアプリケーションはユーザに連続的な体験を提供することができます。</p> <p>本セッションではスピーカーが過去に業務で利用したことのある次の3つのリアルタイム通信基盤を紹介し、それぞれの特徴や得意なこと不得意なことを紹介します。</p> <ul style="list-style-type: none"> •Socket.io (WebSocket) •Firebase Realtime Database •Realm Mobile Platform <p>また、リアルタイムに画面を更新する際のちょっとしたテクニックやTipsも合わせて解説します。</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	@fushiroyama
2016/11/01 12:39:23	Service in Action	<p>AndroidにAPI Level 1から存在するServiceですが、実は非常に奥が深く、使い方次第では意外なほど高度なことが可能です。本セッションではスピーカーが業務で培った知見から特にbindするServiceにフォーカスし、非常に長い処理や結果がいつ返ってくるかわからないような非同期通信などを複数画面をまたぎつつ上手にハンドリングするTips等を紹介しします。</p>	プラットフォーム (Platform)		30分 / 30 minutes	@fushiroyama
2016/11/01 13:58:43	AccessibilityServiceを使ってアプリの可能性を広げよう	<p>通常のアプリ開発時にはほとんど見ないといっても過言ではないAccessibilityService(ユーザー補助機能)。本来はその名の通り老人や盲目の方など視覚にハンデがある方に向けた補助機能ですが、他のアプリの情報が取得できたりそれを活用したりできるため、使用方法次第ではアプリの幅がぐっと広がる技術です。本セッションでは、AccessibilityServiceそのものの話やAccessibilityServiceを使う上での基本的な機能紹介を踏まえた上で、スピーカーが実際に開発したデザイン確認やデバッグのためのツールを紹介し、AccessibilityServiceで出来ること可能性を提示します。</p> <p>構成</p> <ul style="list-style-type: none"> - AccessibilityServiceってなんなの？ - AccessibilityServiceの基本～はじめてのAccessibilityService作り～ - AccessibilityServiceで出来ること、出来ないこと - AccessibilityServiceを使ったアプリ紹介 - 実際に書いてみてハマったポイントなど 	その他 (Other)	AccessibilityService	30分 / 30 minutes	門田福男
2016/11/01 14:22:36	Deep dive into LeakCanary and memory leak patterns	<p>モバイル開発はハードウェアの制約があるため、如何にしてメモリを効率よく使うか、メモリをリークさせないかが鍵になります。このセッションではメモリに焦点を当て、大きく以下の3つを説明します。</p> <ol style="list-style-type: none"> 1. LeakCanaryのメモリーリーク検出の仕組み 2. Androidにおけるメモリーリークのパターン 3. Android Studioでメモリーリークを検知する方法 <p>最初に、LeakCanaryのメモリーリーク検出の仕組みから、GCの基本的な動きを知ることを目標とします。その次に、無名クラスによる観オブジェクトへの参照などの、Androidにおけるメモリーリークの頻出のパターンを説明します。最後に、Android Studioからメモリーリークを如何にして検出するかの方法について解説をします。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	@stsn_jp
2016/11/01 14:29:22	RxJavaで複雑な処理を組む際に気を付けたいポイント	<p>RxJavaで複雑な処理を組む際に気を付けたいこと</p> <p>対象: 初心者以上(RxJavaを触ったことがある方向け)</p> <p>RxJava上を使って1000行弱程度のそこそこ複雑なストリームを実装をした経験から、その際に学んだTipsを紹介します。</p> <p>普通のAndroidプログラミングでは悩まないことでも、Rxのストリーム上で処理の分岐をしたり、ロジックの分割をしようと思うと正しい書き方に悩む事があります。また処理が大きな見通しが悪くなり、例外的なハンドリングが把握出来なくなったり、思った通りの順序で実行されず悩むことも多々あります。このような悩みは抽象的で検索が難しく、自力で解決するにはそれなりの時間が必要でした。このセッションでは私が実際に悩んだ内容をベースとして、以下のポイントに絞って紹介します。</p> <ul style="list-style-type: none"> * 処理の実行順序を厳密に保証するためには気を付けるポイントの紹介 * for文とrxのストリームの違いを理解する * 用途に応じて適切なオペレータを選択するための考え方 * 見通しよく実装するためのテクニック * ロジックを小さな単位に分割して再利用する実装手法の紹介 * 結果を受け取り、複数の処理に分岐させる実装手法の紹介 * 複雑な例外処理の実装するためのいくつかのパターンの紹介 * 2.x系移行の際に気を付けるポイント <p>特別難しい話は出来ませんが、何となく理解できたRxJavaを、自分の思い通りに使いこなせるための機会になれば幸いです。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	kazy(kazuki yoshida)
2016/11/01 15:11:53	オフラインファーストなアプリケーション開発	<p>モバイルアプリケーションを最優先に位置づけるモバイルファーストという考え方が広まってきています。モバイルファーストはさまざまな要素を含むものですが、その中でもオフライン時の使い勝手を考えることはとても重要です。</p> <p>このオフライン時のアプリケーションに着目した概念にオフラインファーストというものがあります。</p> <p>発表ではオフラインファーストとはどのような考え方なのか、また実際にAndroidアプリケーションをオフラインファーストな考えに基づいて開発する上での技術的課題の整理とそれら乗り越えるための具体的な方法をお伝えします。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	zaki50
2016/11/01 15:17:53	日経新聞紙面ビューアーを支える技術	<p>日経新聞は「紙面ビューアー」という新聞の紙面がそのまま閲覧できるビューアーを提供しています。このビューアーでは、紙面全体を載す1枚の大容量画像を30画分という膨大なビットマップを取り扱うので、メモリのオーバフローに気を付けつつ読者が少しでもストレスなくロードするため、また貧弱なAndroid端末でも読めるように「画像の多段階ロード」や「タイトルロード」、「近接する画像優先ロード」など、大量の画像を扱うための様々な知見が詰まっています。</p> <p>また、新聞紙面はポートレート表示時は1面ずつ、ランドスケープ時には2面同時に、なおかつ1面と最終面は片面表示、さらに2面全体を使った2連番広告など... 様々なモードで矛盾なく表示するために見た目以上に複雑な状況を考慮する必要があります。</p> <p>その他にも、紙面自体は1枚の大きな画像であるにも関わらず、各記事単位で矩形状態を保持しており、ユーザのタップでそこだけがハイライトされたり、記事内に登録したキーワードが含まれる場合は色を変えたり等々、非常にチャレンジングなアプリです。</p> <p>本セッションではこのアプリを支えるこれらの様々な技術や、実際のコードを交えつつご紹介したいと思います。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	@fushiroyama

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 15:26:46	Pure Goで作るAndroidアプリ	<p>■ セッションの対象者 Android 中級者 ~ Go 初級者 ~</p> <p>■ 概要 Go Mobileは、GoogleのGoチームがサブプロジェクトとして実装を進めている GoでAndroidやiOSのアプリを作るためのライブラリ・ツール群です。</p> <p>GopherConをはじめとするGoのカンファレンスでも複数のセッションで扱われるなど注目を集めています。前回のDroidKaigi 2016でも「Go MobileでAndroidアプリ開発」というタイトルでお話しました。</p> <p>Go Mobileは、NativeアプリとSDKアプリという2種類の開発手法を提供してきました。前回発表したとおり、今までは、AndroidアプリをすべてGoで書くには、OpenGL ES 2.0 の関数を使って描画する必要がありました。また、GoからJavaやAndroidのAPIを扱うことは難しく、限られた機能しか提供されていませんでした。そのため、簡単なアプリは書くことはできましたが、実際の現場で使えるとはいえないものでした。</p> <p>しかし、Reverse Bindingの機能が追加されたことにより、既存のJavaのクラスへのGoのバインディングを生成することができるようになりました。</p> <p>参考: https://github.com/golang/go/issues/16876</p> <p>既存のJavaのクラスへのバインディングが自動で生成できるようになったため、Androidが提供するクラス群もGoから扱えるようになり、UIのコードも含めPure GoでAndroidアプリを書くことができるようになりました。</p> <p>参考: https://go-review.googlesource.com/#/c/31170/ (まだマージされていませんがPure Goで書かれたアプリのサンプルです)</p> <p>このセッションでは、Reverse Bindingの機能を中心に、Pure GoでAndroidアプリの開発方法について説明致します。また、Go MobileがどうAndroidやJavaの機能をラップ/バインディングしているのかについてもお話す予定です。</p> <p>Go Mobileは、まだまだ実験的なプロジェクトですが、徐々にと足回りが整ってきました。私がGo Mobileのソースコードを覗いて得た最新のGo Mobileの情報をご提供できればと思います。</p> <p>なお、Android開発者向けのお話を予定ですので、Goについてはあまり知らなくても問題ありません。</p> <p>【参考】 ・Go Mobile: https://github.com/golang/mobile ・Go MobileでAndroidアプリ開発 (DroidKaigi 2016 発表資料): http://www.slideshare.net/takuyaueda967/go-mobileandroid ・Go for Mobile Games (GopherCon 2016 発表資料): http://www.slideshare.net/takuyaueda967/go-for-mobile-games</p>	プラットフォーム (Platform)		50分 / 50 minutes	tenntenn
2016/11/01 17:24:18	Androidディスプレイのフラグメンテーションと対策	<p>■ 対象者 初級者向け</p> <p>■ 内容 Androidには様々な画面サイズ、解像度を持った端末が存在します。最近ではスマートフォンの大型化が進み、6.7インチの画面を持つスマートフォンも出てくるようになりました。スマートフォンとタブレットの境界がぼやける中で、アプリはどのようにスマートフォン/タブレットを判別し、端末に最適なレイアウトやリソースを提供していくべきなのでしょう？ 本セッションでは、Androidにおけるディスプレイサイズフラグメンテーションの実態と具体的な対策方法を紹介したいと思います。</p> <p>■ 構成 - Androidの解像度分布 - densityとdpi - dpiによる端末の種類判別 - 代替リソースの提供とレイアウトの切り替え - ベストプラクティス/注意点など - マルチウィンドウ機能への対応 - 画面のズーム(density変更)機能への対応 - spatchの風 - どのdpiに画像を提供すべきか - フォレットはスマートフォン？タブレット？</p>	UI-デザイン (UI and Design)		30分 / 30 minutes	nein37
2016/11/01 17:53:55	ある日突然「AndroidのシステムAPIを利用したアプリを開発してください」と言われたら	<p>■ 対象 Android OSをカスタマイズしたり、AndroidシステムAPIを利用したアプリを開発したい方 初級～中級向け</p> <p>■ 内容 Androidには特別なアプリからしか利用できないシステムAPIがあります。たとえば端末のファクトリーリセットやバックアップ・リストアなどがそうです。通常の開発用SDKでは、これらのAPIを利用したアプリをインストールしたりメンテナンスすることはできません。 これらのAPIには@hideアノテーションが付けられていて、通常利用する開発用SDKには含まれないようになっているからです。 そういったシステムAPIを利用したことが無かったスピーカーが、突然アサインされて悪戦苦闘した経験や、利用時の注意点などをお話したいと思います。</p> <p>■ 構成 権限の違い-通常アプリ、Systemアプリ、Platformアプリ 端末に定義されている権限-一覧を取得する hiddenメソッドを利用するための開発用SDKを作成する hiddenメソッドの利用例 端末のバックアップ・リストアを実行する (他のアプリが前面にいるときに)バックグラウンドから画面の任意の場所をタップする 端末をファクトリーリセットする</p>	プラットフォーム (Platform)		30分 / 30 minutes	Kiyotaka Soranaka
2016/11/01 18:20:59	Androidのクラウドテスト環境まとめ	<p>世の中には数多くの Android 端末が存在し、画面サイズや OS バージョンなどもまちまちで、実機を揃えて動作確認するにも限界があります。Firebase Test Lab や AWS Device Farm などのクラウドテスト環境を使えば、実機を手元に用意しなくても、それらの機種での動作確認やテストを行うことができます。本セッションでは、Android アプリ開発において利用可能な各種クラウドテスト環境の特徴や機能比較を紹介致します。</p>	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	辰濱健一
2016/11/01 18:43:53	モバイル意識の低いチームを中から変えていく話	<p>対象者 - モバイル意識の低い会社/チームに所属している人</p> <p>たとえばWebアプリケーションを主力サービスにしている場合、どうしてもモバイル対応を後手に回しがちです。開発だけではなくCIやアプリのリリース作業の効率化など、やるべきことはたくさんあるはずなのにPC版に注力するあまりモバイル対応に工数を割けなくなるというのはよくある話だと思います。</p> <p>これはなにが原因なのでしょう。社内のモバイル意識の低さ？市場需要のなさ？根本的な技術力不足？原因は様々なと思います。そこで弊社では社内のモバイル意識の向上や開発補助、技術力の底上げ、環境整備などを目的とした「モバイル意識の高いチーム」を立ち上げました。</p> <p>本セッションではモバイル専門チームが1年でどれだけ社内のモバイル事情を変えられたのか、どう変えていったのかを事例を挙げながら紹介します。</p>	その他 (Other)	チームビルディング	30分 / 30 minutes	刈川 陽平

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 18:43:58	Android定期実行処理入門	<p>対象: 初心者以上</p> <p>「n時間おきに○○(任意のタスク)したい」という機能を実装することになった場合、Android SDKが提供しているクラスだけでも幾つか選択肢があります。また、利用するクラスやOSのバージョンによって期待できる精度や条件、振る舞いが異なります。本セッションでは、JobSchedulerの解説を中心として、2017年に定期実装処理を組み込むなら知っておきたい情報を基礎的な部分から紹介したいと思います。</p> <p>現在予定している目次:</p> <ul style="list-style-type: none"> * AlarmManagerを避けたい理由について * JobSchedulerについて * 期待できる精度の件について * firebase-jobdispatcher-androidなどのバックポート系ライブラリについて * その他の選択肢について <ul style="list-style-type: none"> - push通知を利用するなど * アプリアップデート時やOSアップデート時などのそれぞれの挙動について * 動作検証時に確認しておきたい項目について 	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	kazy(kazuki yoshida)
2016/11/01 19:28:50	Kotlinのdelegated propertyを知る	<p>Kotlinの特徴としてnull安全や拡張関数がよく取り上げられるため、影に隠れがちですが便利な機能としてdelegated propertyがあります。</p> <p>本セッションでは、このdelegated propertyの仕組みをJavaのデコンパイル結果から、使い方や応用例をKotterKnifeやKodein、Kotprefといったライブラリ、組み込みのdelegated propertyを通して紹介していきます。</p> <p>■構成</p> <ul style="list-style-type: none"> - delegated propertyとは？ - 組み込みdelegated property - ライブラリでの応用 - delegated propertyの作り方、使い所 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	chibatching
2016/11/01 19:49:37	Getting Started in Android Studio and IntelliJ IDEA Plugin Development	<p>Can make talk 30-50 mins, flexible.</p> <p>So, you've got a great IDE. It's super smart and helpful, and you love it. But there's just that one thing... one simple little thing you wish it did. It would be trivial, really. If only someone would add it.</p> <p>This talk will tell you everything you need to get started developing a plugin for Android Studio / IntelliJ IDEA, including:</p> <ul style="list-style-type: none"> * Getting set up * The basic building blocks of the IDE and of a plugin * Examples to learn from * Testing * Gotchas * Publishing <p>By the time you leave this talk, you'll be ready to dive in and create your own basic plugin, and can avoid some pitfalls along the way.</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Jon Reeve
2016/11/01 19:55:13	Using Git like a pro: "Master-only" workflow, other branching strategies and Git hacks	<p>In this talk we will be discussing advanced Git techniques, how you can utilize different branching strategies to achieve a rapid workflow depending on the structure of your team, followed by an assortment of 'hacks', or useful tricks, that allow you to optimize the way you work and share code with others.</p>	開発環境・ツール (Productivity and Tooling)		50分 / 50 minutes	Savvas Dalkitsis
2016/11/01 20:13:48	セカイフォンを支える技術	<p>セカイフォン(リアルタイム翻訳アプリ)を2010年に初版を公開して以降、現在まで6年以上開発を継続しています。その期間に、様々な課題に遭遇しては対応してきました。具体的には、以下の技術的な取り組みを実施したことで、様々な知見が得られました。</p> <ul style="list-style-type: none"> ・Android最新バージョン対応(1.5の時代から2.x/3.x/4.x/5.x/6.x/7.xと適宜対応) ・新デバイス対応(スマホ/タブレットに加え、ウェアラブル(Android Wear・Google Glass・Sony Smart Eyeglass)/ChromeCast) ・UI/UX改善・新機能実装 ユーザ評価で発生した課題対処や新機能実装をライブラリ等を活用して対応) ・マルチOS対応を考慮したAndroid版開発(iOS版との相互運用のためのDB周りのリファクタリング) <p>これらの中から特に工夫した点を、具体的なエピソードを交えてご紹介したいと思います。</p> <p>※セカイフォン公式サイト http://www.lplatform.org/blog/sekai-phone</p>	その他 (Other)	Platform, Library, UI/UX, Device	30分 / 30 minutes	Kenichi Kambara
2016/11/01 20:34:58	マテリアルデザイン実践再入門	<p>■対象者 初級者 今更マテリアルデザインって何？ときけない人 どこまで対応したらマテリアルデザインを名乗っていいのかわからない人 iOSやWebにもマテリアルデザインは使えるって聞いたけどできっこないと思っている人</p> <p>■概要 Androidアプリ開発では当たり前のようになっているマテリアルデザインですが、マテリアルデザインをアプリのものにすることができているでしょうか。 デザインプレートに従ってなんとなく実装していたり、デザイナー任せにしていたりしないでしょうか。 minneはWeb、iOS、Androidに展開し3年以上続いたサービスです。 すであるサービスでどのようにマテリアルデザインを解釈して取り入れたのか、マテリアルデザインを取り入れたことでアプリ全体の実装にどのように影響したのか、マテリアルデザインを導入する上でハマったことやよかったことなど、minneにマテリアルデザインを導入した経緯を元に、これからマテリアルデザインに対応していきたい人やもっとよくしていきたい人の参考になれば幸いです。</p> <p>マテリアルデザインの基本のおさらい</p> <ul style="list-style-type: none"> ・カラーパレット ・タイポグラフィ ・レイアウト ・コンポーネント Androidでの実装のコツ ・SupportLibraryの賢い使い方 ・Theme&style ・APIv21以前との付き合い方 実際のアプリでの実装例 ・minneのカラーパレットとタイポグラフィ ・アイコンフォント ・コンポーネントの使いどころ ・マルチプラットフォームなサービスでの取り入れ方 <p>SupportLibraryに含まれるコンポーネントの細かい実装のお話はしません。</p>	UI・デザイン (UI and Design)		30分 / 30 minutes	mochico
2016/11/01 20:46:07	Global@one source開発を行うチームの苦悩と工夫とエトセトラ	<p>弊社開発部隊は Tokyo / US / UK に開発拠点をもち、それぞれのregionにアプリをリリースしているのですが、アプリはone sourceで開発し build flavorを分ける事で3種類のアプリを作っています。</p> <p>one sourceとは、つまりgit repositoryが1つで、それを各サイトのメンバーで触っていくという事です。</p> <p>現地にエンジニアを配置することで、各regionにローカライズされたアプリを素早くリリースすることが出来る反面、物理的に遠ましく離れたサイトでのone source開発を行うというのは様々な所に困難が潜んでいます。例えば時差なんかも一つの問題です。</p> <p>このセッションではそういった開発体制でのAndroid開発の実例を紹介し、どのような困難があり、それらをどうやって解決しているのか、ツールや運用方法のご紹介から、one sourceを上手に触っていくための技術的な工夫まで、幅広くご紹介していこうと思います。</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	tsuyoyo
2016/11/01 20:46:58	より速く効率的にサービス開発するための mBaaS のうまい実例	<p>JavaとC/C++で書かれたネイティブライブラリの相互運用は面倒な問題です。Javaの世界では、この問題にさまざまな人たちが取り組み、数多のバインディング自動化機構を構築してきました。本セッションでは、特にAndroidでの利用可能性を意識しながら、一方でJava以外の技術も視野に入つつ、それらの技術のいくつか、実例も含めて紹介したいと思います。(この種のフレームワークはmoving targetなので、当日まで主にどれを紹介するかは未定です。)</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	小山哲志

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 20:56:47	Data binding libraryと始めるMVVMパターン	<p>(対象者)</p> <ul style="list-style-type: none"> Android中級者～ 特に、設計・アーキテクチャに関心のある人 data binding libraryの使い方に関心のある人 <p>(概要)</p> <p>肥大化するAndroidアプリにおいて、どのような設計にするか、どのようなデザインパターンを採用するかは開発者にとって気になることです。これまでMVC、MVPなど様々なデザインパターンが多く、開発者によって試されてきましたが、昨年Googleから公式にdata binding libraryが公開されたことにより、MVVMパターンの注目が高まっています。GitHubなどではすでに多くのMVVMパターンのサンプルが公開されていますが、その多くは画面数が2~5くらいの極めてシンプルなもの、実際に本番で運用しているアプリへ適用できるのか不安な方も多いのではないのでしょうか(私もその一人でした)。</p> <p>本セッションでは、data binding library、RxJava、 Dagger 2を使ったMVVMパターンのアプリ設計について検討します。実際に本番で運用している中規模～大規模なアプリケーションにMVVMを適用していく上での課題などを共有します。</p> <p>(草案)</p> <ul style="list-style-type: none"> MVVMパターンとは ビュー/ビューモデル/モデルの説明 MVVMを採用するメリット、デメリット 既存アプリを移行する上で困ったことの共有 MVVMにおけるテストの書き方 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	kobakei
2016/11/01 21:11:46	Developing Apps for Daydream and Tango	<p>Starting an App with a Virtual or Augmented reality component is a bit daunting for most. When starting out, frameworks like Unity3D or Unreal Engine simplify a great many things.</p> <p>But, from an Android developer perspective, you need to forget all about the nice APIs and libraries you're used to. You'll need to shoehorn your business logic in frameworks that are designed to build games first, not apps -- not a great proposition.</p> <p>Let's explore the path less traveled together! In this session, we'll see how you can build VR and AR applications, in Java and Android Studio, thanks to Google's Daydream and Tango Java SDKs, plus a few helpful 3D graphics libraries.</p>	プラットフォーム (Platform)	Tango / DayDream	50分 / 50 minutes	Etienne Caron
2016/11/01 21:22:13	多言語化アプリの開発とTipsいろいろ	<p>対象者</p> <ul style="list-style-type: none"> 多言語化をこれから始めようとしている開発者 多言語アプリを公開している開発者 初心者～上級者 <p>言語</p> <ul style="list-style-type: none"> スライドは英語 トークは日本語 <p>内容</p> <ul style="list-style-type: none"> 多言語対応を始めるには 対応する言語の選定方法 多言語化のための準備 開発プロセスに多言語化を取り入れる 翻訳の管理方法 管理システムの紹介 いろいろTips CIとの連携 古いAPIでも使える日付フォーマットの生成方法 翻訳文がTextViewから流れないようにする方法 翻訳者にコンテキストをより多く与える方法 おまけ: ユーザと一緒にアプリを翻訳する方法 	その他 (Other)	多言語化	30分 / 30 minutes	João Orui
2016/11/01 23:51:34	RxJava for Beginners	<p>■ 対象者</p> <ul style="list-style-type: none"> RxJavaに興味はあるが使ったことがない人 <p>■ 概要</p> <p>RxJavaに興味はあるが、メリット・デメリットがよく分からず導入に踏み切れていない方もおられると思います。本発表はそんな方の背中を押すものになります。RxJavaとは何なのかというところから始まり、RxJavaが解決しようとしている課題とRxJavaを導入することによる弊害、RxJavaのユースケースを紹介したいと思います。</p> <p>■ 目次</p> <ul style="list-style-type: none"> リアクティブプログラミングとは RxJavaの紹介 RxJavaが解決しようとしている課題 RxJavaを導入することによる弊害 RxJavaのユースケース Async TaskやAsyncTaskLoaderの置き換え 非同期処理の待ち合わせ 入力フォームのリリクエション 他にもいくつか追加予定 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	yuyakaido
2016/11/01 23:51:19	Unit testing with Robolectric	<p>■ 対象者</p> <ul style="list-style-type: none"> これからテストを書こうと思っている人 Robolectricに興味はあるが使ったことがない人 UIテストの実行時間が長すぎて困っている人 <p>■ 概要</p> <p>Android開発におけるソフトウェアテストは、JUnitを使った古典的なユニットテストからEspressoを使ったUIテストまで様々ありますが、本発表ではRobolectricを使ったユニットテストに焦点を当てて、Robolectricの導入方法から発表者が実務で培った実践的な知見をお伝えします。</p> <p>■ 目次</p> <ul style="list-style-type: none"> なぜテストを書くのか テスト手法 ユニットテスト UIテスト Robolectricの紹介 モックライブラリの紹介 Robolectric 基礎編 Robolectricの導入方法 簡単なサンプルテスト モックオブジェクトを使ったテスト Robolectric 応用編 疑似UIテスト 権力ユニットテストでカバーするためのテクニック おまけ CI上での実行 Kotlinとの連携 	メンテナンス (Quality and Sustainability)		50分 / 50 minutes	yuyakaido

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 23:51:06	Google Dagger for Beginners	<p>■ 対象者</p> <ul style="list-style-type: none"> Dependency Injectionに興味はあるがよく分かってない人 Google Dagger初心者 <p>■ 概要</p> <p>ここ最近ではMVPやMVVM、Clean Architectureといった様々なアーキテクチャの話題が多く取り上げられていますが、それらのアーキテクチャ全てに共通する要素としてDI (Dependency Injection) があり、DIを使わずにこれらのアーキテクチャを実装するのはメリットが半減してしまいます。様々なアーキテクチャや理論を言えば、1つのクラスに持たせる責務を1つに限定して、クラス同士を疎結合にしたいということで、このクラス同士の依存関係をより疎結合にするためにDIが使われます。本発表では、なぜDIを使うのかということから始まり、発表者が実務の中で培ったGoogle Daggerのナレッジを全てお伝えします。</p> <p>■ 目次</p> <ul style="list-style-type: none"> DIとは なぜDIを使うのか アーキテクチャとDIの関係性 DIコンテナの紹介 RoboGuice Square Dagger Google Dagger Google Dagger 基礎編 Module Component Inject Google Dagger 応用編 Scope Subcomponent 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	yuyakaido
2016/11/01 23:49:15	Error Handling in RxJava	<p>■ 対象者</p> <ul style="list-style-type: none"> RxJava初級者～中級者 <p>■ 概要</p> <p>Android開発にRxJavaを取り入れる事例がかなり増えてきており、Androidの標準APIのイケてない部分を解消したり、複雑な非同期処理をうまく記述したり、といった知見はかなり充実してきていると思います。一方で、RxJavaを使う場合のエラーハンドリングに関する知見はそれほど広まっていないのではないかと感じます。そこで本発表では、エラーハンドリングにおけるいくつかのパターンを取り上げて、RxJavaを使った実装方法を紹介したいと思います。</p> <p>■ 目次</p> <ul style="list-style-type: none"> RxJavaのおさらい RxMarblesについて エラーハンドリングで使うOperatorの紹介 <ul style="list-style-type: none"> onErrorReturn onErrorResumeNext retry retryWhen よくあるエラーハンドリングの実装方法 <ul style="list-style-type: none"> Toast/SnackBarを表示する 代わりにデータを表示する リトライする 少し複雑なエラーハンドリングの実装方法 <ul style="list-style-type: none"> リトライするかどうかをユーザーに委ねる エラーの種類によって異なるハンドリングを行う 他にもいくつか追加予定 	設計・開発手法 (Architecture and Development Process/Methodology)		50分 / 50 minutes	yuyakaido
2016/11/01 22:43:56	Android Security 最新情報！！	<p>■ 対象者</p> <ul style="list-style-type: none"> Android 初級者～中級者 <p>■ 内容</p> <p>Android N になり、様々なSecurity関連のアップデートがありました。DirectBoot, KeyAttestation, NetworkSecurityConfig, ScopedDirectoryなど。本セッションでは、AndroidSecurityの観点で、APIの紹介やその使用方法から、今のAndroidアプリセキュリティのベストプラクティスを考察します。また、Framework内でのアップデート、SelinuxやSignature関連も紹介します。</p> <p>■ 構成</p> <ul style="list-style-type: none"> Android Security の概要 Android N Security APIの紹介 実際の開発でのノウハウ Android Framework の最新Security事情 <p>■ 本セッションに関連する話題</p> <ul style="list-style-type: none"> Android Security AndroidFramework 	メンテナンス (Quality and Sustainability)		30分 / 30 minutes	Naoki Yano
2016/11/01 23:06:42	Dressing Up with Drawables	<p>Drawables are a fundamental component to the styling and rendering of Android UIs. Even if you are just starting Android development, you probably used drawables at least a little to add images or style widgets inside of your layouts. The drawable system is also a great mechanism for doing highly custom graphics, states, and animations in Android views.</p> <p>This talk will provide a look at the different ways to use Drawables, including a quick review of built-in drawables and how to add them, an example of how to use drawable states in views, as well as an example of building custom drawables.</p> <p>対象者 ・RecyclerViewを使ったことがある人</p>	UI・デザイン (UI and Design)		30分 / 30 minutes	Huyen Tue Dao
2016/11/01 23:17:12	LayoutManagerをつくらう	<p>RecyclerViewの要素をどう並べるかを決めるLayoutManager。Support LibraryではLinearLayout, Grid, StaggeredGridの3つが用意されているが、もっと凝ったレイアウトや複雑なレイアウトをしたい場合、自分でLayoutManagerを作るしかない。この発表ではいくつかのサンプルレイアウトを使い、自作LayoutManagerを作るときのコツや注意点を紹介する。</p> <p>発表内容(案)(予定)</p> <ul style="list-style-type: none"> LayoutManagerとは何か RecyclerViewとLayoutManagerの関係 LayoutManagerの動作仕組み 簡単な自作LayoutManagerを用意し、実際のコードを見ながらカスタマイズの要点を紹介 まとめ 	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	consomme72
2016/11/01 23:18:08	EventBus Basics	<p>In this session I'm gonna talk about Event bus, what is that, where to use, how to implement and some practical tips. Note that this talk won't refer to entire architecture of application, it's just about technical stuff and pragmatic patterns. I hope listeners will get an overview about Event Bus in 30 minutes.</p> <p>Target:</p> <ul style="list-style-type: none"> Beginner-Intermediate level <p>Contents:</p> <ul style="list-style-type: none"> What's Event bus ? (5min) Event Aggregator pattern Pub/Sub pattern It's not a magic -What's going on under the hood- Where to use? <ul style="list-style-type: none"> Don't abuse! Just use it to decouple layers Event bus in Android(10min) <ul style="list-style-type: none"> square/otto is deprecated! greenrobot/eventbus RxJava Pros/Cons of each libraries and how to select Broadcast receiver/ Callback Listener suck ? Practical Tips(10min) <ul style="list-style-type: none"> Notification among layers like Fragment/Activity, Service/(View)Controller Get the state of app is foreground/background Deal with FCM/GCM appropriately Reactive programming vs Event bus Q&A(5min) 	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	hotchemi

bett	セッション名 / Talk title	概要 / Abstract	トピック / Topic	トピック(その他) / Topic(Other)	発表時間 / Talk slot	セッション用氏名 / Name
2016/11/01 23:32:09	つらいけど楽しい、初心者がAndroid開発を好きになるまで	<p># 対象 初心者～(Android開発者なら誰でも)</p> <p>Androidを始めたときに挫折しかけた経験がありませんか？(そのときの自分を忘れていませんか？) Androidを始めて半年程度(&Webのプログラミングも1年未満)のわたしが、業務で新規機能開発を担当した際の経験を中心に話したいなと思います。</p> <p>具体的には、どのようにAndroid開発を始めたか、という初期の話から、実際に開発して頂いた、FragmentのライフサイクルやFragmentManagerのバックスタックの仕組み、指摘されるけど気づけ無いNullチェック、リスナーによるイベントの委譲やコールバック処理、サーバとの同期/非同期通信など、についての経験談を話したいと思います。</p> <p>自分と同じようにAndroidを始めて間もない方にとってはAndroidの開発をどのように学んでいくか、つまづきやすいポイントはどこかなどの情報を紹介できるとと思います。また中・上級者の方でも、例えばNew Androiderが入ってきたときのアドバイスの参考の役にたてばと思います。</p>	その他 (Other)	Android開発全般	30分 / 30 minutes	taihaku
2016/11/01 23:37:33	2つのアプリ、1つの設計のデザイン指針	<p>【対象者】 デザイナーの考えを知りたい ワンソースプロダクトの設計に興味がある デザイナーが1人、もしくは少人数のチーム</p> <p>【概要】 Quipper社では、スタディサプリとQuipperアプリという2つのプロダクトを持っています。全て一つのソースコードから生まれています。</p> <p>さらに、教育系という特性上、iOSとAndroidでほぼ同時に設計を行うことが多いです。</p> <p>ワンソースから生まれるプロダクトで、デザイナーがどのような考えで日々設計に動いているか、ビジネス側の理解と開発の整合性、エンジニアの方々との協力関係はどうなっているのかなどお話ししたいと思います。 ※デザイナー目線でお話するので、プログラムの話は一切出てきません。</p> <p>【目次】 - ワンソースプロダクトの良さ - 日本と国外の設計の違い - iOSとAndroid - それぞれのOSの特性を理解する - マテリアルデザインで動かした感想</p>	設計・開発手法 (Architecture and Development Process/Methodology)		30分 / 30 minutes	meycy
2016/11/01 23:37:55	エラーと戦うためのデバッグ法	<p>対象: 初心者以上</p> <p>効率的に開発を進めるにあたり、欠かせないと言えるのがデバッグ作業かと思えます。コード上ではエラー表示がなかったとしても、いざビルドを行えばエラーログが出力され、インストールに過ぎずけてもアプリを動かさず不具合が起き得ます。修正をするためには、その問題が何故どうして起きたのか、原因を特定する必要があります。</p> <p>Androidの開発では、開発環境のAndroidStudioに強力なデバッガが備わっています。また、各種ライブラリなどを利用して、様々な角度からアプリの状態をチェックし、開発をより効率的に進めることができます。</p> <p>このセッションでは、Android開発をはじめたばかりの初心者を想定し、エラーを解決するための糸口として、Android開発で使えるデバッグ方法について紹介を行うことができると思っています。具体的には、AndroidStudioでのデバッグ方法やライブラリの紹介などを行う予定です。</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	山崎亮
2016/11/01 23:53:41	アプリの質を上げるには、ガイドラインの作り方	<p>【対象】 アプリのデザインをもっと良いものにしたい方 デザイナーとの連携がうまく行っていない気がする方 デザイナーを過さず、スムーズに開発してみたい方</p> <p>アプリデザインの質を上げたい、でもデザイナーが一人ではできない。 【スタディサプリ】の質を上げるために、エンジニアと協業して、デザインガイドラインを作成、適応しました。 エンジニアとデザイナーの良いコラボレーションを生み出すことができました。 実際に行った「アプリの質を上げるためのガイドラインフロー」をご紹介します。</p> <p>【目次】 - モックとなんかちがう問題 - 荒れた時代 - ガイドラインをつくって困ったこと - 昔の思ってるガイドライン - ガイドラインを適応するのに必要なこと - 得意な人を探す - 実際のフロー - その後</p>	UI・デザイン (UI and Design)		30分 / 30 minutes	meycy
2016/11/01 23:57:16	リリース自動化と効率のよいリリースフローを求めて	<p>Androidアプリのリリース自動化を導入したことによるメリットをお話したいと思っています。 対象者: 特にレベルは問わないです。リリース作業が面倒だと感じている人にはおすすめできると思っています。</p> <p>リリース自動化に関する知見は一部の人は持っていますが、実際に実行に移したり導入に際しての障壁などを経験していらっしゃる方は少ないようにも思えます。</p> <p>大企業・スタートアップを問わず、リリース自動化によって受けられる恩恵は大きいのでその知見を共有することで、自動化についての知見が増えれば、と思っております。</p> <p>また、自社で開発しているアプリは国内では珍しいほどに海外比率が高いため、途上国向け開発をする上でなげリリース自動化が非常に有効だったのかをお伝えできればと思っております。</p> <p>トーク内容: リリース自動化に使うツールの紹介 実際に使っているツールとそのツールを選んだ理由 導入方法を詳細に説明 実際にアプリの配信を自動で行っている 自社リリースフローの紹介 どういったことが大幅に解消されたかの紹介 多言語化対応時やAndroidアプリを途上国向けに展開するうえでリリースの自動化が生きた場面の紹介 などについて発表させていただきたいと思っています。</p>	開発環境・ツール (Productivity and Tooling)		30分 / 30 minutes	Ryo Sakaguchi