

「ちょっとだけ変なことをしている会社＝技術変態アルタスファイブ！」です。

日頃より、ご支持ご愛顧を賜り、誠にありがとうございます。今年も新たに、技術に対する情熱を持った新人 3 人を迎え、組織は 22 人となり、一段と活気に溢れています。

今後とも、技術力の向上を目指し、開発力を高めてまいります。

それでは、2023 年の上半期について振り返り、アルタスファイブの取り組みについてご紹介させていただきます。

今号の記事のテーマは、世界的にも話題となっている「AI」と「サステナビリティ」です。ただし、これだけだと内容が被りそうなので、他のテーマもいくつか加えました。私たちの取り組みと考え方について詳述いたします。

### 目次

|    |                                |    |
|----|--------------------------------|----|
| 01 | ナレッジの共有に GPT を使用する             | 小池 |
| 02 | システム開発によるサステナビリティへの貢献について知ったこと | 雲居 |
| 03 | Angular プロジェクトチーム内でのコード共有会     | 沢辺 |
| 04 | Spring Boot で構築したアプリのリファクタリング  | 西岡 |
| 05 | メンタリングへの挑戦と成長                  | 磯貝 |
| 06 | ChatGPT 以前と以後では仕事のやり方が変わる      | 佐藤 |
| 07 | お菓子紹介                          | 宇野 |

## 01 ナレッジの共有に GPT を使用する

### 開発部：小池

昨年 ChatGPT3.5 のリリース以来、生成 AI を実際のビジネスに利用するトレンドが続いています。

現在、私が取り組んでいるプロジェクト企業様でも AI 利用が強く推奨されており、その環境を提供いただいています。

強力であってもまだまだ未成熟なこの技術が、保守的な日本社会であっても瞬く間に浸透しつつある理由は为什么呢？この会話する AI に Web から触れる分には賢そうでもあり、酷くポンコツにも見えます。

そこで初学ながら業務の視点から ChatGPT の API に触れることとしました。

予てよりナレッジ共有は課題となっており、また参考資料が多く Web 上に存在することから「カスタマイズした情報を GPT に検索させる」実装を試みてみました。

実装は Python、OpenAI API に加え、LangChain を用いて機能を拡張していきます。

「カスタマイズした情報」の保存先としては、実際の利用シーンで想定できるデータストアとして Elasticsearch を選択しました。

まず、気がついたのは実装に必要となるほとんどの基本機能が、高レベルに提供されていることでした。

データストアは VectorStore と呼ばれるもので、高次データの類似度から対象のレコードを検索できる能力が要求されます。

このためには事前にベクトル化した情報も保存する必要がありますのですが、この実装は langchain.vectorstores.ElasticVectorSearch にインデクシングするドキュメントと Embeddings(つまりベクトル化手法)を渡すだけとなっています。

もちろん Embeddings もあらかじめ提供されているので、それを引数に渡すだけです。

面倒なデータの投入も各種 Loader が提供済みで、テキスト、CSV、Word、Excel はもちろん、S3 や GCS のようなクラウド

ドストレージの Loader も当然のように存在します。

Youtube の Loader までもあるのですから、こんなことで手は煩わせないぞという強い意志が感じられます。

回答に関しては、ユーザーからの入力値と類似度が高いドキュメントを VectorStore から取得(Retrieval)して、そのドキュメントを GPT のプロンプトとして入力することで、GPT に適切な回答を生成させます。

この辺りも 3 メソッドくらいを呼び出すだけで実装可能です。

ただ Retrieval の段階で適切なドキュメントが取得できていないといけないわけで、この辺りは泥臭い工夫が必要なようです。

流石にこの簡易な実装ではナレッジ検索として実用には届きませんでしたが、それでも尤もらしく検索できてしまうのが今時の AI らしくありました。

実装ベースで調べることで LangChain の機能に「これは知識共有目的ではとても収まらないな」と思い始め、周辺技術を爆発的に発展させる GPT のポテンシャルには慄然とし、WebUI からは見えなかった激流のような力を感じることができました。

今更ではありますが「ChatGPT ヤバイ」と巷の流行に乗れそうな気が少しだけ致します。

#### 編集からの一言：

ChatGPT は履歴を保持する設定にしてあると、その履歴が学習に使われてしまうそうなので、履歴を残さない設定で使うと、機密漏洩を防げるようです。また API は、そもそも履歴がないので、API を使うのも、1 つのやり方だと思います。使いこなして生産性を上げるのが得策ですね。（開発部 佐藤）

## 02

### システム開発によるサステナビリティへの貢献について知ったこと

#### 開発部 雲居

芸人のとにかく明るい安村がイギリスの某オーディション番組でハネた動画を見て挑戦してみないとわからないことや笑いの力、日本人の誇りなどを感じてちょっと感動してしまった開発部の雲居です。

皆さんはサステナビリティについて考えたことはあるでしょうか？

昨今 SDGs(Sustainable Development Goals（持続可能な開

発目標）)という言葉が浸透してきており、聞いたことがある人も多いと思います。私も聞いたことはある程度の認識でした。そんな中今年の 4 月、弊社佐藤の新入社員への挨拶の中で「サステナビリティ」という言葉があり、たまたま聞いていた私はシステム開発でサステナビリティに関係するのかと思っておりましたが、少し気になりサステナビリティについて改めて調べ自身が携わった案件もサステナビリティに貢献しているものがあるのかと振り返ったところ当てはまるものがいくつかありました。

そこで今回はシステム開発によってサステナビリティに貢献できることについて述べさせていただきます。

まずは私自身が関わった案件の中で気づいた物を挙げたいと思います。

一つ目はマンションの管理員をシステムに置き換える案件になります。こちらは入居者による問い合わせや管理業務を人の代わりに行うチャットボット及び管理システム開発の案件になります。このシステムによってサステナビリティへの貢献として自動化や遠隔監視により、管理業務の効率化や的確で迅速な問題解決が実現できるため、利用者の利便性や満足度が向上することが考えられます。これにより快適なユーザーエクスペリエンスを提供することはサステナビリティの一環になると思われます。また、24 時間のシステム対応により管理員のスペースや電力の消費削減によって人件費やエネルギーの削減にもつながると考えます。

二つ目は工事現場向けカメラアプリの案件になります。こちらは工事現場で施工状況の写真を撮る際に電子黒板を用いるシステム開発の案件になります。このアプリを用いることにより現場で実際に黒板を使用する必要がなくなることになり、木材や塗料などの資源を節約し、廃棄物を削減する効果が期待できると考えます。また、広義に捉えると黒板の製造・輸送に伴う CO2 排出を削減することができることも考えられます。これ等はサステナビリティに関連すると思います。

私自身が携わった案件を振り返り IT システムによるサステナビリティへの貢献はあることを知ることができました。そこで他にも何かないかと思ったところ様々考えられることですが、少しシステムによるサステナビリティへの貢献になると考えられることを上げたいと思います。

デジタル文書管理システムの導入やワークフローのデジタル化により、紙(印刷物)の使用を削減し、ペーパーレスを実現することができると考えられます。これにより、広義には森林の保護や廃棄物の削減に貢献できると考えます。また、クラウドコンピューティングと仮想化技術を組み合わせることで、データセンターやサーバーの数を削減し、エネルギー消費を削減することができると考えられます。こちらは、必要なときにリソースを拡張することで、無駄な資源の使用を防ぐこと

もできると考えます。

このように私自身が携わった案件やサステナビリティ目線で思案したことは傍から見ると大したことはないように思える貢献かもしれません。しかしながら塵も積もればなんとやらではありませんが、一社会人として自身の仕事貢献していると思うとモチベーションや仕事の喜びにもつながると思います。

サステナビリティへの貢献と考えると大げさになってしまいかも知れませんが、ITシステムの役割として忘れずにおこうと思いました。

今回はサステナビリティの目線で振り返りましたが、これに関わらず自身の携わる案件がどう貢献しているかを知ることや考えることはとても大事なことで今回この記事を書くにあたり改めて認識いたしました。

これを読んでいただいた方も何かの気付きに繋がれば良いなと思いながら今回はここで締めさせていただきます。

#### 編集からの一言：

弊社が取り組んでいるSDGsは、目標8の「働きがいも経済成長も」です。会社も社員も持続できるためには、やりがい、報酬、人間関係のすべてが整備されている必要があるということで、人事コンサルの会社に入ってもらって、給与体系や評価制度の見直しを行っています。それから、人間関係という点については、心理的安全性を勉強しながら、高めていこうとしています。そのあたりは、また別の機会に、記事としてご紹介致します。（開発部 佐藤）

03

## Angular プロジェクトチーム内でのコード共有会

### 開発部 沢辺

いつもこの時期になるととうもろこしを食べるというお話をしていましたが、今年は全然とうもろこしをあまり食べていません。

今年は子供のじゃがいもほりで家にじゃがいもが大量にあるので、ほぼ毎日じゃがいも料理を食べています。ジャーマンポテトがお気に入りです。

今回は「チームの知識共有の方法」がテーマということで、私の担当している Angular プロジェクト内での取り組みについてお話させていただきます。

前回に私が「開発スピードをUPさせること」をテーマにした際のお話にも、共通コンポーネントなどを wiki にまとめたことで開発スピードがあがるように意識しているとい

うことをお話ししました。今回のお話はこれに少し関連したお話になります。

元々、一部の共通機能の作成などは私が担当していたものや、前任者が担当して私に引き継いだものなどが多く、wiki にまとめてはいたのですが他のメンバーは理解できていないものが多いというのが実際のところでした。

機能によっては属人化してしまってると感じていたところもありましたので、チーム内で月1のコード共有会を開催することにしました。

wiki にまとめていたものは、wiki の内容を見返して整理をし、wiki にまとめていなかったものは wiki にまとめるところから始めて、とにかく共有会では wiki をベースに説明を行いました。

wiki に拘っている理由は単純に、後で見返せるものがあった方がよいと思ったからです。

共有会で説明しても100%理解するのは難しいと思いますので、共有した機能の改修をチームメンバーが担当する際に wiki を見て思い出しながら改修してくれることを期待しています。

こういった活動を昨年の10月頃から始め、今は一通り共有したいと考えていた機能の共有は終わりました。

共有会をやってみて、チームメンバーからは共有会によって理解できたことはかなりあったという声もありましたので、少なからずやった意味はありました。

今後はコードに関わらず複雑な仕様をしている機能の共有などをしていこうと考えていますが、まだまだ改善点はたくさんあると思っています。

例えば、共有会を開催した後に何を共有をしたのかの記録をどこにも残していませんでしたので、これはどこかに残さないといけないと。

wiki をベースにしているのであれば、共有会の内容をまとめたページがあった方がよいのは明らかですね。

他にも今回は私の方で説明をしましたが、チームメンバーに共有会を開催してもらうことも考えています。

チームメンバーには負担がかかってしまうので多少迷いもありますが、チームメンバーにとっては聞いているだけより理解度などは深まると思っています。

まずはこの辺りをチームメンバーに相談して更なるチーム開発の改善に繋がってほしいです。

そして、共有会のようなものがあると、リモートワークがメインとなっている寂しがり屋の私にとっては、誰かと会話する機会が増えて少しうれしく感じるものでありました。

#### 編集からの一言：

社内で勉強会を催すとき、勉強会のスライドを作成する主催者が一番勉強することになるのですが、同じことが当てはまるのだと思います。自分が作成した資料は、なんとなく愛

着も出てきたりするし、書き溜まってくると、達成感もありますから、この取り組みは、方々に波及していくとよいと思います。（開発部 佐藤）

## 04

# Spring Boot で構築したアプリのリファクタリング

### 開発部 西岡

少し前に、SF 小説の三体を読み終わりました。読み始めたのは一年以上前なのですが、スターウォーズや、スタートレックのような宇宙を舞台にした、スペースオペラ的な物語が好きな自分にとって、第一部の内容があまり好みではなく、大分放置してしまいました。しかし、第二部以降はハード SF 小説という内容で、4 日で黒暗森林、死神永生を読み終わってしまいました。Netflix でドラマ化もするらしく、人力 x86 マシンの映像が見られるとのことで、楽しみにしています。

さて、今回のテーマはコード品質とその維持です。私は現在、Spring Boot を使用したプロジェクトに取り組んでいます。このプロジェクトでは、各ページでデータの取得処理が記述されているため、取得方法に変更があると、処理の修正とテストをすべてのページで行わなければなりませんでした。

そのため、機能追加時にこの問題に対処するための修正を行うことになりました。今回はこの機会に試したリファクタリングの内容を説明します。検索フォームの仕様変更があり、各ページで共通の挙動で表示される検索フォームを実装の共通化しようとしていました。

まず、対象の検索条件コンポーネントを切り分けて、Thymeleaf でのビュー定義を fragment として別ファイルに移動しました。そして、利用する側のページのビュー定義を、該当のコンポーネントの配置箇所に replace で挿入するように変更しました。また、各画面のフォームクラスでも、画面ごとに同じ検索条件の実装が存在していました。このため、同じ目的のプロパティや処理がすべてのページのフォームクラスに記述されていました。この問題に対処するため、コンポーネント用のサービスクラスとフォームクラス用のインターフェースを作成しました。そして、ページの検索条件コンポーネントの入力値を取得する場合には、該当のコンポーネントのインターフェースをフォームクラスに継承し、データの受け渡しには該当のインターフェースからアクセスするような実装を行いました。

これらの変更により、画面ごとに散在していたサービスクラスの実装箇所をまとめることができました。

私は今回初めて Spring のプロジェクトに触れましたが、このフレームワークは長い間使用されているため、情報が古くなっているものも多くありました。そのため、モダンな方法についての知識を得ることに苦労しました。また、今回の案件

では Java8 を想定した環境でしたが、var キーワードを使用できない点や、interface で private な関数を扱えないなど、困る状況もありました。しかし、Java 周りの状況から、今後も Java8 を使用する機会があると思うので、Java8 での対応方法についても覚えておく必要があると感じました。

### 編集からの一言：

Java8 + 少し世代の古い Spring の組み合わせは、まだまだ、現役で使われているところがたくさんあります。技術的負債は、大きいコストになる前に、少しずつ解消したいです。Web で検索しても、最新世代の技術記事がヒットするので、生産性も、低くなっていきます。でも、この問題は、ユーザーの理解を得るのが、なかなか大変です。良い方法があれば、私も教えてほしいです。（開発部 佐藤）

## 05

# メンタリングへの挑戦と成長

### 開発部 磯貝

最近、暑くなってきましたね。いつの間にかセミの鳴き声も聞こえ始め、夏の訪れを感じます。

弊社では昨年度からメンター制度を導入し、OJT とは別に新入社員のサポートを行う取り組みを行っており、私は今年度の新入社員のメンターを担当させていただくことになりました。

メンターになってまだ1か月ほどですが、思ったことを書いてみようと思います。

そもそも、メンターとは、メンティーの仕事やキャリアの手本となり、若手や未熟者に対し、メンタル面を含めた仕事の悩みの相談者のことを指します。社会人としてのあり方や仕事に対する考え方など、幅広い視点からメンティーの成長を支援する役割が求められます。

こう見ると、なかなか難しそうです。

私自身も入社2年目で、昨年まではメンティーの立場でした。そのため、キャリアの手本となれるほどの実績も持っていないし、社会人としてのあり方や仕事に対する考え方をアドバイスできるほどの経験もまだありません（これについては、私がもっと努力しなければなりません）。

そんな私がメンターとしてできることは何だろうかと考えたときに、とことん相手の立場に立って考えることなのではないかと思いました。メンティーは、学生からいきなり社会人

としての生活が始まり、右も左も分からない状態で日々を過ごしていると思います。私も2年目だからこそ、そのような不安や悩みを理解しやすいはず。まずはメンティーの良き理解者となり、悩みや不安があれば自分事として考えてみる、その上で解決への道筋をメンティーと一緒に見つけ出していくことが、私がメンターとしてできることのひとつだと思います。

メンタリングを行う上で、コミュニケーションは欠かせません。しかし、リモートワークという都合上、メンティーとのコミュニケーションはSlackなどのオンラインコミュニケーションツールを使って行うことになるため、対面よりもコミュニケーションが希薄になりがちです。この点についても工夫が必要だと感じています。現在は週に数回、Web会議ツールを使ってメンタリング面談を行っていますが、それ以外にもチャットなどを活用して、意識的にコミュニケーションを取ることを心掛けるようにしています。なるべく一人で不安を抱え込む時間が短くなるようにしたいと思っています。

また、これまでメンターをやってみて分かったことは、メンター側もメンタリングを通して得られるものが多いということです。コミュニケーションスキルやコーチングスキルの向上はもちろん、メンティーからの質問や意見に対して考えることで、自身の経験や知識を再評価し、新たな学びや気づきを得るきっかけにもなります。

これからも、より良いメンターとしての在り方を試行錯誤しながら、メンティーが成長していけるように精一杯サポートしていきたいと思っています。

#### 編集からの一言：

弊社の場合、組織が小さいため、メンターとメンティーの仕事の距離が近いです。本来は、他部署の人とペアが作れると、もっと気楽に相談しやすくなると思うのですが、部署が1つしかないの、なかなか思うようにはいきません。弊社にあった制度にするためのアレンジが必要で、月1でメンターの振り返りを行って、弊社流メンター制度を作っています。（開発部 佐藤）

## 06 ChatGPT 以前と以後では仕事のやり方が変わる

### 開発部 佐藤

ChatGPT にプロレスやろうよ！と大喜利を仕掛けたら、私が提示したキャラで、プロレスをシミュレーションしてくれると言うんです。

リングネーム：胸板アツシ

スタイル：ルチャリブレ

必殺技：念仏

結果は、審判がカウントを4まで数えたので、審判の反則で無効試合となって終了。

バカな技を繰り返したら、ChatGPT ってば、レフェリーがカウントを4まで数える筋書を出してきたという、なんとお洒落なことを。

一本取られました。←これオチです。ChatGPT より弱いかな。（笑）

さて、本題です。

それにしても、ChatGPT の対話のレベルには、まったく以って驚きました。

時間を見つけては、システム開発の仕事にどのように活かせるか、試しています。

いくつかは妄想も含まれますが、ご紹介します。

### （試行1）ゼロベースで始める ChatGPT とのペアプロ

私は、これまでReactを使ったプログラムを書いたことがないので、マニュアルやGoogleを使わないと、1行も書き進められないのだけど、試しに、この知識レベルで、マニュアルとGoogleを封印して、ChatGPT だけを使って、どこまで作れるのかやってみました。

結果、なんと、Reactのプログラムが作れて、期待どおりに動きました。

作業中は、まさに、ChatGPT とペアプロしている感じです。ほぼ無知の状態から始めると、結構、時間はかかるものの、Reactの開発方法も、それなりに身につきました。いやはや、すごいです。

ペアプロって、長時間やると疲れます。作業に疲れるというよりは、人に疲れます。でも、ChatGPT とのペアプロは、その手の疲れは無いので、ストレスのないペアプロができます。

今までだったら、マニュアル読んで、チュートリアルをやって、Reactを少し理解してから、本来の開発作業を始めるところですが、ChatGPT とのペアプロ開発は、そんなまどろっこしいことをやらずに、いきなり本丸の作業に取り掛かれます。これまでにない、仕事のやり方です。

そして、何を作りたいのか、どういう機能にしたいのかをプロンプトで与えるので、このテキストが仕様書として使えることも、やってみて気づきました。

うまく使いこなすと、確実に生産性があがります。

## （試行2）ストレスなユニットテスト

他の誰かが作成したユニットテストのコードって、メンテしづらいけど ChatGPT は1つの解決策になるかもしれません。プロンプトで本体プログラムを与えて、パラメタライズドテストを作って！と依頼すると、しっかり作ってくれます。

そして、パラメータの組み合わせの意図をコメントで出力するように指示しておけば、パターンの網羅性を確認できて、人間が書いたコードよりも、読みやすいかも！と思いました。

パターンも、最初から、結構、網羅されていました。モックを使うように指示すると、モックも使ってくれます。

ここからは、妄想なのですが、これは、ユニットテストを半自動にできるのではないかと、想像が広がります。

ユニットテストを実行して、エラーになったところを ChatGPT が修正して、これをエラーが無くなるまで繰り返したら、カバレッジを計測して、実行されていないコード部分を指定して、ChatGPT にその部分のテストを追加するように指示して、再度テストを実行して・・・といった一連のワークフローの自動化です。

実現すると、ユニットテストの作業のストレスが軽減されそうです。GitHub の Copilot を AutoGPT 的に動かす感じの、そんなツールです。

ChatGPT 周辺のツールは世界中でものすごい勢いで開発されているので、たぶん、誰かが作っているように思います。

## （試行3）賢い実装パターン

本体ロジックも、デザインパターンを使ってロジックを考えてくれ！と指示すると、しっかりデザインパターンを使って実装してくれます。

使うデザインパターンは指定しないと、作ってくれないので、デザパタがなんであるかは、指示する側が知ってないといけないけど、ああ、そんな実装のやり方があったか！と勉強になることもあります。

## （試行4）だるい実装作業の巻き取り

DTO みたいなものも、嫌な顔しないで、作ってくれるし、テストデータも作ってくれる。だるい系の作業は、ChatGPT にやってもらうとよいでしょう。

DB のテーブルから、Entity を generate してくれるツールとかは、ORM に付属しているけど、ChatGPT でそこに機能を付加できるような気がします。（これは妄想です）

テストデータは、Faker が作ってくれるけど、ChatGPT だって作ってくれます。そして、たぶん、ChatGPT の方が楽な気がします。（これも試してないので妄想です）

## （試行5）難易度の高い設計

DDD のような、ちょっと難易度の高いモデリングをやらせてみると、これまたビックリ。

私自信、DDD を完全に理解しているか？というと、そこまで完全ではないので、迷いながらモデリングすることになります。書籍にも、開発が進む中でブラッシュアップしていくものだとあります。

でも、なんだろうか、ChatGPT の場合、あんたは、エリックエバンスなのか！と言いたいくらいに、一発で洗練された印象のモデルを提示してくれました。

なぜ、そういうモデルになったのかの説明を求めると、自分の中での理解が”あやふや”だったことに気づかされ、これまた勉強になりました。

難易度の高いものほど ChatGPT が真価を発揮するのかもしれませんが。

## （試行6）もっとも得意なのは上流工程

これも検証出来ていないのだけど、おそらく、システム開発の各工程の中でも、一番、力を発揮しそうなのが、上流工程かもしれません。

ChatGPT は文章を生成する AI なので、戦略提案とか、要件定義のような上流工程の仕事は、相性がよいと思います。

業界知識がなくても、ChatGPT に聞けば教えてくれます。そんなことも知らないのか？と嫌みを言われることもないです。

調べたいテーマを段階的に掘り下げながら、壁打ちを繰り返すと、いろんな知識を獲得できます。

一回聞いたくらいでは頭に入っていないことも多くあるので、何度も聞けるのは、ありがたいです。

時間無制限にユーザーと打ち合わせ機会を設けることはできないので、ChatGPT が代わりになるなら、効率が良いです。

そのためにも、あらかじめ、プロンプトを用意しておく、上流工程はナレッジが共有されるだろうと想像できました。

たった今は、長文を処理できないので、工夫が必要ですが、近い将来、制限なく長文を処理できるようになったら、劇的に効率化されるかもです。

## （まとめ）

ユーザー企業が ChatGPT を使い始めると、上流工程のナレッジはコモディティ化していくことになって、SI 事業の中でも付加価値の高かった上流工程の仕事の何%かは、効率化される可能性があります。

日本ディープラーニング協会（JDLA）の YouTube 動画では、これからの開発の仕事は、プロンプトエンジニアリングになっていくと語っていました。また、別の動画では、すでに、コンサルティングの仕事の 70% は、ChatGPT で置き換え可能であると示唆していました。

しかし、これが問題だというわけではなく、逆に人間だけが可能な業務に専念でき、より質の高い仕事が行えるようになると述べられていました。

仕事のやり方が変わっていきます。

そして、その変化は大きな速度で進行していると、JDLA の理事である東大の松尾教授が指摘しています。

ChatGPT は、使い方のガイドラインを制定して、早く、使いこなす側に立った方がよさそうです。

## 07 お菓子紹介

管理部 宇野

今期も、日頃の感謝の気持ちを込めまして、社内で食べ比べしたお菓子をご紹介します。

最近は狙っているお菓子が売り切れていることが多く、これを食べてみたいという思いに執着すると、本当に欲しい時に手に入らないのでは？という自分との葛藤が生まれます。

買うことができないというのは、これは売り切れるほど美味しくて人気なのでは？という思考になり、気になって気になって…

記録はあるのに、調べたら潰れている…なんてことがご時世的にもよくある世の中なので、こういったお菓子があり続けること事態が嬉しいことですよね。

こうして、日々お菓子を探し続けるネットサーフィンは続いたのであった。

さて、今期のおすすめお菓子は 3 種ございます。

### 薫製菓子店 薫るサンド



こちらは京都にある薫製お菓子の専門店です。

パリパリ食感の薫製チョコレートをとろりとした薫製ピーナッツバターではさみ、さらにそれを分厚いクッキーではさんだサンドです。

す。

チョコレートとピーナッツバターが薫製されたものですが、全体的に薫製の香りが強すぎることはなく、スモークの香りが苦手な方でも食べれると思います。薫製よりも分厚いクッキーの香ばしさがすごいです。そしてボロボロになる系クッキーです。うまい棒並みにボロボロします。チョコが甘すぎず、ピーナッツバターがいい感じの甘みと香りがあり、クッキーの香ばしさがいいマリージュしています。

薫製感はあまり感じられませんが、シンプルにバランスが良く美味しいサンドです。

### NEWYORK PERFECT CHEESE-ニューヨーク パーフェクトチーズ-



これは知っている方も多いのではないのでしょうか？溶けてしまうため夏は避けていた…もとい、買えないので味見が遅れてしまいました。いや、まだまだ並びますね。就業前に買いに行きました

が、開店前から列ができていましたよ！

「チーズ好きのためのチーズ好きによる極上お菓子」という売り文句ですが、後味にチーズ独特の香りが残るわけではないので、比較的に食べやすいチーズお菓子だと思います。

ゴーダチーズを練りこんだラングドシャ生地には程よい甘みがあり、ふわふわのミルククリームとデンマーク産のチェダーチーズ入りホワイトチョコレートは、軽い食感でチョコもチーズも主張が強すぎず、サイズの的にも食べやすい一品です。溶けやすいので暑い日の外での持ち運びには気を付けましょう！

## バトンドール



こちらは関西にしか売って  
いません。関東で手に入れ  
るにはグリコの公式で買う  
しか…

「プール・クラリフィエ」  
と呼ばれる黄金色の”澄まし  
バター”がおいしさの秘密だ  
そうです。左記画像の黄金

色の液体が”澄ましバター”だそうです。バターを溶かし余計な  
ものを取り除いたもの、画像だけでも美味しそうですね。高  
カロリーはおいしい。こちらと発酵バターを混ぜたダブルバ  
ター製法という独自の製法が！

シュガーバターの見た目は高級なプリッツ。食べてみるとプ  
リッツと表現したことを謝りたくなる高級バターの風味が！  
とても甘いわけでも、バターがくどいわけでもない、香りと  
食感を楽しむお菓子。

もう一度、高カロリーはおいしい。

こちらはこだわりの強いお菓子なので、賞味期限が少し短  
め。ここは注意ポイントです。

今回は買いにくいお菓子が多いですね。関東にはなかった  
り、とても並んだり午前中で売り切れたり…しかし、これを  
読んで気になってしまったそこのあなた！

是非食べてみてください。

### 編集からの一言：

頭を使うと糖分が欲しくなります。どうせ食べるなら、おい  
しい方がいい。おやつ探しを始めて、彼は、試食したお菓子が  
98個となりました。けっこう食べました。食レボコメント  
も、溜まってくると、楽しくなります。（開発部 佐藤）

次号につづく