

# ITエンジニアという仕事

三輪 智樹

LINEヤフー株式会社

# 今日のゴール

---

## 学科・進路選びのヒントを持ち帰る

OBのリアルな経験を共有する

- 大学でどう過ごすといいかイメージできる
- 進路選択の判断材料が増える
- ITエンジニアという選択肢を知る

# まず教えてください

---

## プログラミング経験ある人？

授業でも趣味でも

# まず教えてください

---

## エンジニアに興味ある人？

なんとなくでOK

# まず教えてください

---

行きたい学科がすでに決まっている人？

手を挙げてみてください

# 今日の話は

---

情報系に進む人以外にも聴いてほしい

学科・進路選びに役立つヒントをお伝えします

# 今日の流れ

---

## 1. 自己紹介

どうやってエンジニアになったのか

## 2. ITの仕事

エンジニアにも種類がある

## 3. 技術を覗いてみよう

LINEの裏側

## 4. 大学時代の過ごし方

エンジニアになるまで

## 5. AI時代のエンジニア

これからどうなる？

## 6. まとめ

今日お話ししたこと

# 1

---

## 自己紹介

どうやってエンジニアになったのか

# プロフィール

---

## 基本情報

- **出身:** 早稲田実業学校（2018卒）
- **大学:** 早稲田大学  
基幹理工学部 情報理工学科
- **現在:**  
LINEヤフー株式会社 エンジニア

## やっていること

- 金融の部署でエンジニア
- 全社の生成AI活用Project

# 高校時代の自分

---

- 高1: 高等部から入学。陸上部に所属。勉強は普通（テストは平均点くらい）

- 高2: 暗記が苦手だったため理系一択

- 高3: 父の仕事の影響で機械系に興味 → 基幹理工学部 学系2を選択

友人がプログラミングやっていたが、全く興味がなかった

# 学部1年での転機

---

## 大学1年のC言語授業

- 何も分からない...
- 友人に課題を助けてもらう
- スラスラ解く友人がかっこいい



## このままじゃヤバい

- 長期休みで必死に勉強
- 少しずつ理解できるように
- プログラミングの面白さに気づく

# 学部2年で情報理工学科へ進学

- 少しできるようになったただけだったが、情報理工学科へ進学を決意
- 競技プログラミング: アルゴリズムの面白さに気づく
- アルバイトの誘い: 授業＋独学で力をつける

入学当初は機械系志望だったが、気づいたら情報系に夢中だった

# インターンで実務経験

---

## 学部3年～大学院

- インターン・業務委託を開始
- 合計10社で実務経験
- 様々な業界・技術に触れる

## 気づいたこと

- 授業と実務は全然違う
- 実際に使われるものを作る緊張感
- やって見ないとわからない

# なぜエンジニアを選んだか

---

- 1 インターンで実務の面白さを知った
- 2 10社の経験で「これを仕事にしたい」と確信
- 3 学び続ければ成長し続けられる

ポイント: 実際にやってみて「合う」と思えた

# 1章のまとめ

---

明確な目標がなくても、何があるかわからない

- ✓ 高校時代、エンジニアなんて考えてなかった
- ✓ 大学で出会って、ハマって、進路が変わった
- ✓ 特に学部1年は大切な時期

# 2

---

## OBが語る、ITエンジニアのリアル

### エンジニアにも種類がある

# IT業界の全体像

---

企画



設計



開発



テスト



運用

エンジニアは「開発」だけじゃない。全部に関わる

# グループワーク

---

「LINE」を作るなら？

どんな役割の人が必要だと思う？

# グループワーク

---

## ルール

- グループで話し合い（3分）
- 最低3つの役割を考える
- 「なぜ必要か」も考えてみよう

思いついたものを自由に出してみよう！

# みんなが考えた役割

---

「何を作るか決める人」「デザインする人」「プログラムを書く人」...

そういう役割、**実際に存在します**

それぞれの役割に名前がついている。

# プロダクトマネージャー (PM)

---

## 何を作るか決める人

- 「既読機能、つける？つけない？」
- 「スタンプは有料にする？無料にする？」
- 「次はどの機能を優先して作る？」

チームの方向性を決める、いわば「船長」のような存在

# デザイナー

---

## どう見せるか決める人

- トーク画面のレイアウトは？
- 吹き出しの色や形は？
- スタンプの大きさは？

「見た目」だけでなく「使いやすさ」も考える

# フロントエンド / モバイルエンジニア

---

## 画面を作る人

- モバイルエンジニア: スマホアプリの画面を作る
- フロントエンドエンジニア: Webブラウザの画面を作る

みんながLINEを開いて見ている画面、この人たちが作ってる

# バックエンドエンジニア

---

## 裏側の処理を作る人

- 送信ボタンを押したらメッセージが届く
- 相手が読んだら「既読」がつく
- 過去のトーク履歴を保存する

目に見えないけど、一番重要な部分かもしれない

# インフラエンジニア

---

## 土台を作る人

- LINEは日本だけで何千万人も使っている
- 全員が同時にメッセージを送っても落ちない
- データを安全に保存する

サーバーというコンピュータを管理して、サービスの土台を支える

# QAエンジニア

---

## 品質を守る人

- メッセージが届かない...
- 既読がつかない...
- アプリが突然落ちる...

こういうバグを見つけて報告する。みんなが安心して使えるのはこの人たちのおかげ

# 実は、これはほんの一部

---



## データエンジニア

PMが判断するためのデータを集めて分析する



## 機械学習エンジニア

「おすすめスタンプ」などAI機能を作る



## プラットフォームエンジニア

他のエンジニアが使う道具や環境を作る

# 作るものが変われば、必要な人も変わる

---

## ゲームを作るなら？

- 3Dエンジニア
- サウンドデザイナー
- ゲームプランナー

## 自動運転を作るなら？

- 画像認識エンジニア
- センサーエンジニア
- シミュレーションエンジニア

ITの仕事は本当にいろいろある

## 2章のまとめ

---

- 1 1つのサービスを作るには、たくさんの役割が必要
- 2 今日紹介したのは代表的な例。実際はもっと多い
- 3 自分に合った役割が、きっとどこかにある

# 3

---

## 技術を覗いてみよう

### Webページが表示されるまでの裏側

# 質問です

---

ブラウザでURLを入力してEnterを押したあと  
何が起きてると思う？

# 考えてみてほしいこと

---

## どこに繋がる？

URLからどうやって  
目的地を見つける？

## 何を取得する？

テキスト？画像？  
どんなデータ？

## なぜ速い？

世界中のサイトが  
すぐ表示される理由

## なぜ安全？

パスワードが  
盗まれない仕組みは？

# グループワーク

---

Webページが表示されるまでの流れを  
想像して図にしてみよう

# グループワーク

---

## ルール

- 正確さより**想像力優先**！
- 矢印で流れを書くだけでOK
- 「心配ポイント」も入れてみて（遅い、盗み見、サーバー落ち など）

**時間: 5分**

# 何グループか発表してもらいます

---

# どんな図になった？

# 実際の流れ（簡略版）

---

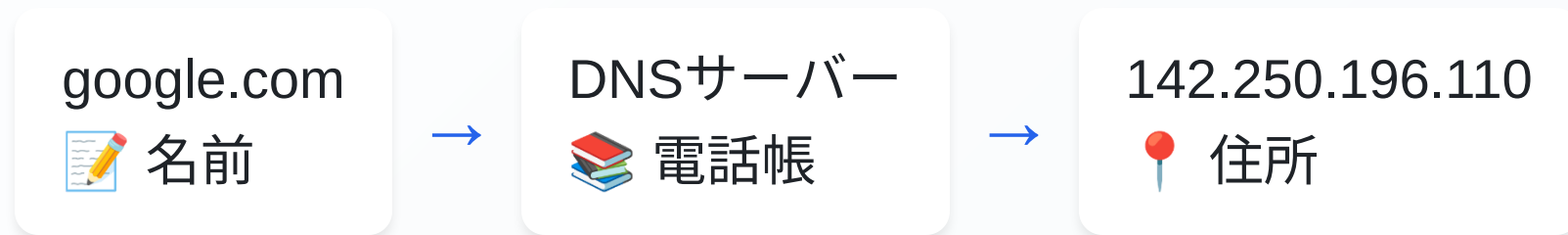


URL → IPアドレス → サーバー → データ取得 → 表示

# すごいポイント①：DNS

## インターネットの電話帳

「google.com」を「142.250.196.110」に変換する仕組み



# すごいポイント②：速度の工夫

---

## 普通に考えると

- 日本→アメリカのサーバー
- 毎回遠くまで取りに行く
- 時間かかりそう...




## 実際は

- CDN（世界中にコピー配置）
- キャッシュ（一度取ったら保存）
- 高速に表示される


# すごいポイント③：セキュリティ

## HTTPS（暗号化通信）

あなたとWebサイトの間の通信を暗号化。途中で盗み見できない

あなた  
 暗号化



インターネット  
 読めない



サーバー  
 復号化

# こんなことも考えてる

---



## 表示の最適化

画像を後から読み込んで、文字を先に表示



## レスポンス対応

スマホでもPCでも見やすく自動調整



## 負荷分散

大量アクセスでも落ちないように複数サーバーで対応

# エンジニアが考えること

---

- 1 大量のアクセスでも動く設計
- 2 どこからでも速く表示する仕組み
- 3 セキュリティと利便性のバランス
- 4 様々なデバイスへの対応

# 3章のまとめ

---

「Enterキー」を押した瞬間から  
膨大な処理が動いている

- ✓ 普段見ているWebページの裏側は超複雑
- ✓ でも、それを作る人たちがいる
- ✓ 技術の積み重ねが「当たり前」を支えている

# 4

---

## 大学時代の過ごし方

### エンジニアになるまで

# 大学6年間でやったこと

---

- 学部1年: C言語の授業だけ + 陸上サークルに少し参加
- 学部2年: 情報系に進学 + 個人開発 + 競技プログラミング
- 学部3年: インターンで実務経験開始 + 大学院進学を検討
- 学部4年: 卒業研究 + 大学院進学 + 業務委託で仕事開始
- 修士1年: 研究 + 就活 + 業務委託継続

# 学業・研究

---

## 授業で学んだこと

- ✓ C言語
- ✓ アルゴリズムとデータ構造
- ✓ ソフトウェア工学

## 研究で得たもの

- ✓ 一つのテーマを深掘りする力
- ✓ 論文を読む・書く経験
- ✓ プレゼン・発表の機会
- ✓ 先生・先輩からのフィードバック

# 授業以外でやったこと

---

インターン・業務委託

実務経験を積んだ

趣味（競プロ）

AtCoderをやっていた

# 就活・進路選択

---

1

自己分析

何が好き？何が得意？

2

企業研究

どんな会社がある？

3

インターン

実際に体験してみる

4

選考・内定

面接・技術テスト

# エンジニア就活の特徴

---

## 他の職種より差別化が明確



### コーディング試験

コードが書ける・書けないがはっきりわかる



### 成果物が証拠になる

自分の作ったもの・携わった仕事が面接で語れる

# 振り返って思うこと

---

## やってよかった

- とにかく何か作ってみた
- インターンに参加した
- 興味のある分野を深掘りした

## もっとやればよかった

- 英語の勉強
- チーム開発の経験
- 異分野との交流

# 高校生の今からできること

---

- 1 興味を見つける: 色々触って「面白い！」を探す
- 2 基礎を固める: 数学・英語・国語は全部使う
- 3 何か作る: 小さくてもいいから完成させる
- 4 発信する: ブログ・SNS・GitHubなど

# 4章のまとめ

---

## 大学は「やりたいこと」を見つける場所

授業だけでなく、自分から動くことが大事

- ✓ 学業と課外活動のバランスが重要
- ✓ インターンは早めに経験しておく と 有利
- ✓ 高校生の今からでも準備できることがある

# 5

---

## AI時代のエンジニア

これからどうなる？

# AIでコードが書ける時代

---

# 2025

AIがコードを書く時代が本格化

「エンジニアいらなくなるんじゃない？」

# 実際に使ってみると...

---

## AIができること

- コードの下書き
- バグの発見
- ドキュメント作成
- 定型作業の自動化



## 人間がやること

- 何を作るか決める
- ユーザーを理解する
- チームをまとめる
- 責任を持つ

# AIで変わること・変わらないこと

---

## 変わること

- 単純作業が自動化
- 開発スピードが上がる
- 一人でできることが増える

## 変わらないこと

- 問題を見つける力
- 人の気持ちを理解する
- 最終判断と責任

# 実はこれ、前にもあった

---

- 1990年代: Excelで経理がいらなくなる？ → ならなかった
- 2000年代: ネットで営業がいらなくなる？ → ならなかった
- 2010年代: スマホで色々な仕事がなくなる？ → 変化した
- 2020年代: AIでエンジニアがいらなくなる？ → ???

# むしろ強くなれる

---

AIが優秀になるほど  
「雑に頼む人」が弱くなる

AIに「ちゃんと仕事させられる人」が勝つ

何をどう作りたいか明確に伝える力が重要に

# これから求められる力

---

## 問題発見力

「何が問題か」を見つける

## 言語化力

考えを正確に伝える

## 判断力

AIの出力を評価する

## 倫理観

技術の使い方を考える

# 高校生の今できること

---

- 1 論理的に考える練習（数学・情報）
- 2 言語化する練習（国語・英語）
- 3 何か作ってみる（プログラミング・ものづくり）
- 4 AIに触ってみる（ChatGPT等）

# 5章のまとめ

---

## AIは敵じゃない、道具

使いこなす側になろう

- ✓ AIで仕事がなくなるのではなく変わる
- ✓ 「考える力」「伝える力」がより重要に
- ✓ 今の勉強は全部つながっている

# 6

---

まとめ

今日お話ししたこと

# 振り返り

---

- 1 自分の話: エンジニアへの道は一つじゃない
- 2 ITの仕事: チームで協力してサービスを作る
- 3 技術の深さ: 「当たり前」の裏に膨大な技術
- 4 大学時代: 今から準備できることがある
- 5 AI時代: 道具として使いこなす側に

# 最後に伝えたいこと

---

「今わからなくても大丈夫。  
やってみたら意外とできる。  
興味を持ったら、まず触ってみて。」

# Q&A

なんでも聞いてください！

# ありがとうございました！

質問があればいつでも連絡ください

@tamofplease | [mmiwatomoki@gmail.com](mailto:mmiwatomoki@gmail.com)