

[ホーム](#) [タイムライン](#) [トレンド](#) [質問](#) [公式イベント](#) [公式コラム](#) [キャリア・転職](#) NEW [Orga](#)

エンジニアのキャリア／人生全体の「サクセス」に徹底して向き合う！Findyの「ユーザーサクセ



この記事は最終更新日から1年以上が経過しています。

@Altech (Sohei Takeno)

## 技術導入についての覚書

アーキテクチャ 技術選定

最終更新日 2022年01月30日 投稿日 2022年01月30日

先日、『[技術を的に当てる技術について](#)』という話をした。これに関連して、影響範囲のある技術を途中から導入するときこのあたりは一定抑えるのが大事なあとと思ってるポイントがあるので、現状認識のスナップショットとしてちょっとまとめておく<sup>1</sup>。

- 最初に確認したいこと<sup>2</sup>
  - その技術はどのような課題を解決しようとしているか？それはどのように要素分解できるか？
  - その技術を入れることで確実に解決される“重要な課題”はあるか？
- 初動
  - その価値が利用の増大に応じて複利的に利くようなものである場合、最初の根を張る必要がある
  - 早い段階で価値を感じられるように、具体性の高い問題解決を行うのが定石
  - 技術を整備したり使う人が理解するには一定のコストがかかる
  - このコストをリターンがちょっとでも上回るようになることを最短で設計するということが重要
- 初動を終えた後
  - そこまでの手応えを振り返り、進むか撤退するかを決める
  - 定常的なワークフローに自然に組み込む、それに困った時に自然と使うように仕向けるなど拡大するようにする
  - ここまで来るとこのリターンとコストの差分であるプラスが増えていく状態になる



20



12

...

- 経過を観察しながら、ボトルネックがあれば対処し、またさらに技術を使いやすくするなどの再投資を適宜行い、開発を加速させていく

## 競合技術がある場合

競合技術がある場合はそれと比較すると良い。これは、より良い選択肢を選ぶという意味もちろんあるが、比較することで元の技術がより良く理解できるというになるためである。

- その技術の競合技術はあるか？その競合技術はどのように要素分解できるか？
- 要素のうち重複する部分と、そうでない部分を明確にする
- それらの要素はなぜ重複して、なぜ重複しないのかを明らかにする
- 特にエコシステムの発展度合いに起因する要素と、そうではない固定的な要素の違いに注意を払う
- この検討の過程で、それぞれの技術がどのような問題設定を敷いており、どのようなスコープの問題解決を行おうとしているのかの理解が深まる
- そのような問題設定はある程度、一般的に存在する問題であることが期待される
- その一般的な問題が自分たちの組織・ビジネス・プロダクトの状況にどの程度適合しているかを検討する
- 適合するのであれば、その技術が現在提供している価値だけでなく、発展することにより将来的に提供される価値も享受できることが期待できる
- 一般的に、技術の仕様と実装だけでなく、その技術が解決しようとしている問題の方を正しく捉えることが非常に重要
  - 参考：[技術を的に当てる技術について](#)
- 性質を理解した上で複数の技術を使いたい場合はポジショニングを明確にする
  - 参考：[Wantedly における Go 導入にまつわる技術背景](#)

## 耐用年数

時間の観点についても触れておく。技術の価値は、それが適切に機能している限りにおいて、入れてから使い終えるまでのトータルの時間積分になる。

このため、外的変化と内的変化のそれぞれについて粗くても良いので見立てを持っておく方が良い。

- その技術の耐用年数はどの程度か？より良い選択肢が市場に生まれるのにどのくらいかかるか？



- 導入後：その選択肢がどの程度、力を持つかを継続的にモニタリングしておくことで、状況が変化した場合に早い段階でキャッチする
- その技術の問題設定が自分たちの状況に適合しなくなるとしたらどのような要因がありうるか？
  - よくある観点：開発者の人数、ビジネスのモデル、サービスの性質、ソフトウェアの複雑性

これらを見立てると次のようなことを考えることになる。

- 技術の耐用年数があまりにも短ければ、モノとしてそっちの方が良いとしてもスルーした方が良い可能性がある
- 一方で、耐用年数内に導入コストを回収できるかは、いかに早く技術を立ち上げられるかでもある
- 立ち上げが早ければ、耐用年数が短くても使った方が良い可能性がある

おまけ：

こうやって見ただけでも事は複雑に絡み合っているわけだが、そういう中で一定のバランスを見てやっていくという動的性質がソフトウェアエンジニアリングの奥深いところなのではないかと思う。また、その意味ではここに書いたことも（あるいは世の中にある重厚な技術選定のフレームワークですら）、ルールではなく一つの目安でしかないので注意したい。

1. 経験上、そういうポイントをちゃんと抑えておけば、「技術を入れたけどなんか良くならなかったね」みたいにな状況になることをかなりの割合で避けることができたり、同じ期間で受けられる恩恵を大きく増やすことができる。 ↩
2. 技術のコンテキストと、自分たちのコンテキスト（困りごとなど）の両面から見るのが重要。自分たちの困りごとばかりに目を向けていると技術の価値を矮小化してしまう可能性があるし、技術の側だけから見ていては使いたいだけ、になってしまいう可能性があるため。 ↩

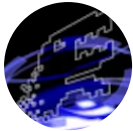


0



20

12



@Altech (Sohei Takeno)

Wantedly という会社でプロダクト開発やソフトウェア設計をしています。人や物の仕組みを知り作っていくことが全般的に好きです。

フォロー

新発売

新発売

ヨドバシ・ドット・コム

新発売	新発売	新発売
¥144,800	¥184,800	¥114,800
新発売	新発売	新発売
¥84,800	¥358,800	¥268,800

ソフマップ

🔗 今日のトレンド記事



@DEmodoriGatsuO (Gatsuo De'modori)  
2024年05月18日

Power Apps & GPT-4oを使って超高速で画像解析アプリを作る！

20

12

♡ 37



@naoya\_347 (なおや)

2024年05月18日

## useSession ? なにそれおいしいの？

React Next.js useSession

♡ 25



@kaku3

2024年05月18日

## AIは仕事ではなく仕事力を奪う？

ポエム 教育 AI pm 新人プログラマ応援

♡ 15



@mkt\_hanada (Makoto Hanada)

2024年05月18日

## 【AI品質・AIテストまとめ】AIシステムの品質を高めるために

テスト AI データサイエンス 品質

♡ 17



@sanjushi003

2024年05月17日

## VMware Fusion 環境 (macOS) に Windows Server 2022 仮想マシンを作成してみた

Mac Broadcom vmware VMwareFusion WindowsServer

♡ 7

[トレンド一覧を見る](#)

### 関連記事 Recommended by



#### 問題駆動型プログラミング学習法

by Sicut\_study



20

12



## IPA試験問題不備(令和6年春期ネットワークスペシャリスト午後2)

by angel\_p\_57



## VSCodeで"kawaii"開発環境を作ろう

by Crysta1221



## GPT-4o の画像認識能力がすごい！カロリー推定アプリを作ってみる

by sakasegawa

## 30代におすすめ！低糖・低脂肪&1本53kcal注目の●●

PR 雪印メグミルク株式会社

## 人感センサーを活用したスマート商品棚など。日立の「公開型生体認証基盤」活用事例

PR 株式会社 日立製作所

## コメント

この記事にコメントはありません。



コメントする



プレビュー

コミュニティガイドラインに基づき、良識ある内容を心がけましょう。

テキストを入力

0B / 100MB

投稿する

## 記事投稿キャンペーン開催中



20

12



## アクセシビリティの知見を発信しよう！

2024/05/07~2024/05/31

[詳細を見る](#)



## 音声認識APIを使ってみよう！

2024/04/10~2024/05/21

[詳細を見る](#)

[すべて見る](#) ➔

How developers code is here.

© 2011-2024 Qiita Inc.

ガイドとヘルプ

[About](#)

コンテンツ

[リリースノート](#)

SNS

[Qiita（キータ）公式](#)



20

12

<a href="#">プライバシーポリシー</a>	<a href="#">公式コラム</a>	<a href="#">Qiita 人気の投稿</a>
<a href="#">ガイドライン</a>	<a href="#">アドベントカレンダー</a>	<a href="#">Qiita（キータ）公式</a>
<a href="#">デザインガイドライン</a>	<a href="#">Qiita 表彰プログラム</a>	
<a href="#">ご意見</a>	<a href="#">API</a>	
<a href="#">ヘルプ</a>		
<a href="#">広告掲載</a>		

Qiita 関連サービス	運営
<a href="#">Qiita Team</a>	<a href="#">運営会社</a>
<a href="#">Qiita Jobs</a>	<a href="#">採用情報</a>
<a href="#">Qiita Zine</a>	<a href="#">Qiita Blog</a>
<a href="#">Qiita 公式ショップ</a>	

