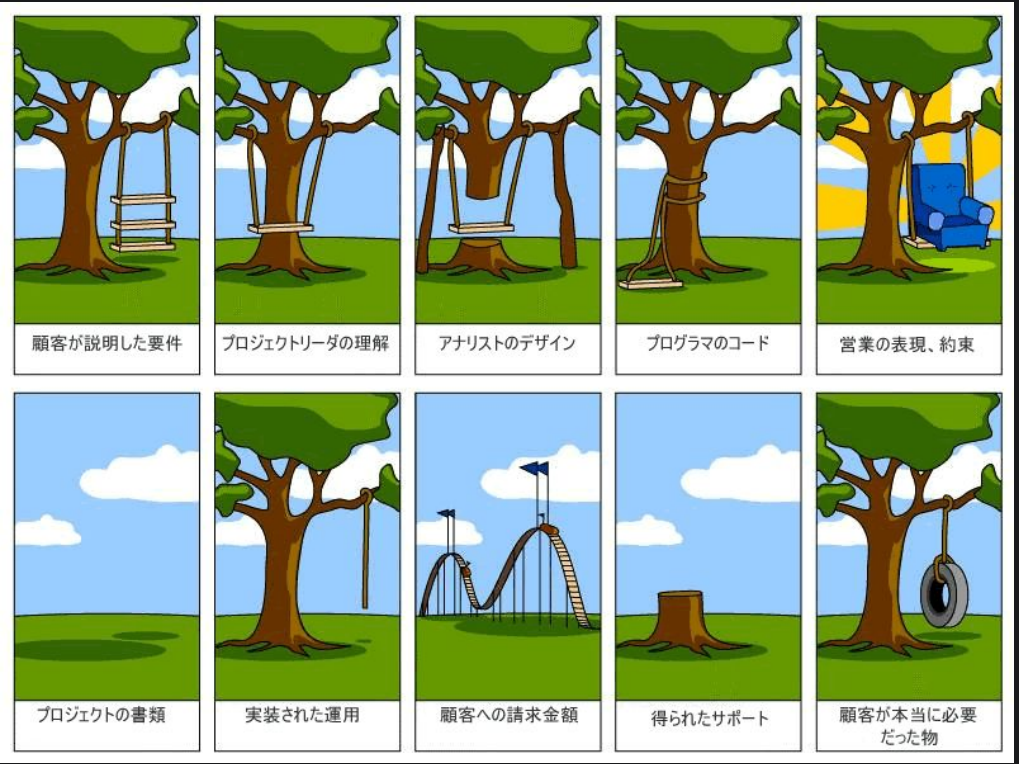
顧客が本当に必要だったもの

– パフォーマンス改善編

【ハイブリッド開催】わたし史上、最高のチューニング

What is it?


顧客が説明した要件
プロジェクトリーダの理解
アナリストのデザイン
プログラマのコード
営業の表現、約束
プロジェクトの書類
実装された運用
顧客への請求金額
得られたサポート
顧客が本当に必要だった物

What is it?

結論：仕様を正しく変更するのが

一番パフォーマンスに効く

What is it?


ポストする
そーだい@初代ALF
@soudai1025
プロモーションする
仕様を変えるのが一番負荷対策になるんだよなぁ #pgcon20j
午後2:50 · 2020年11月13日
ポストのエンゲージメントを表示
7

What is it?

ずっと同じことを言ってる

What is it?

ずっと同じことを言ってる
↓
今日はもうちょっと具体例の話

あじぇんだ

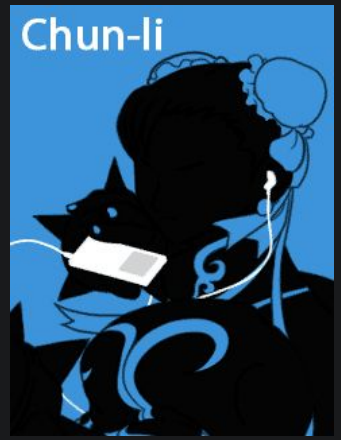
1. 自己紹介
2. 複雑な仕様がRDBを殺す
3. 特効薬：仕様を削る
4. まとめ

あじぇんだ

1. 自己紹介
2. 複雑な仕様がRDBを殺す
3. 特効薬：仕様を削る
4. まとめ

自己紹介

曽根 壮大（40歳）
そ　ね　　たけ　とも



Have Fun Tech LLC 代表社員
株式会社リンケージ CTO 兼 COO

- 日本PostgreSQLユーザ会 勉強会分科会 担当
- 3人の子供がいます（長女、次女、長男）
- 技術的にはWeb/LL言語/RDBMSが好きです
- コミュニティが好き

あじぇんだ

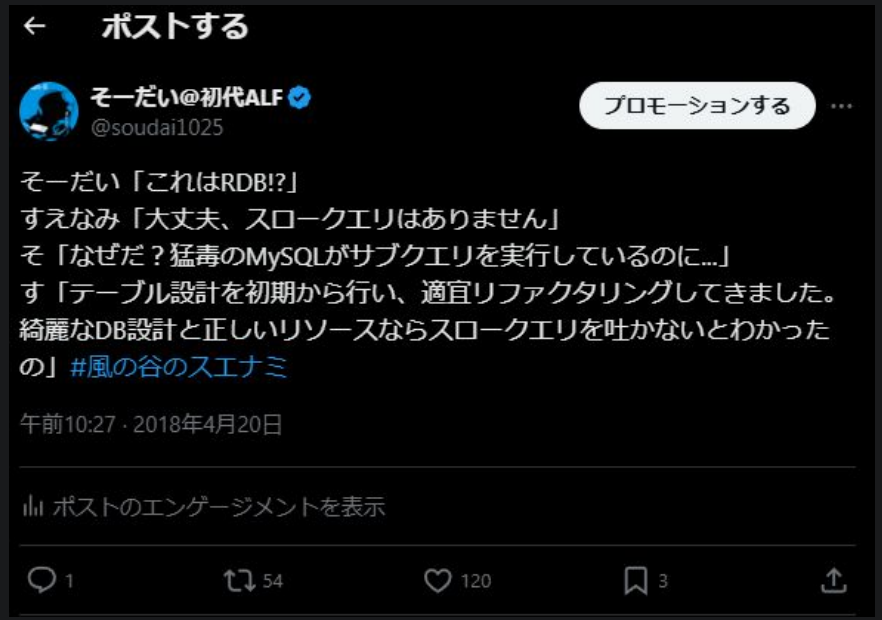
1. 自己紹介
2. 複雑な仕様がRDBを殺す
3. 特効薬：仕様を削る
4. まとめ

複雑な仕様がRDBを殺す

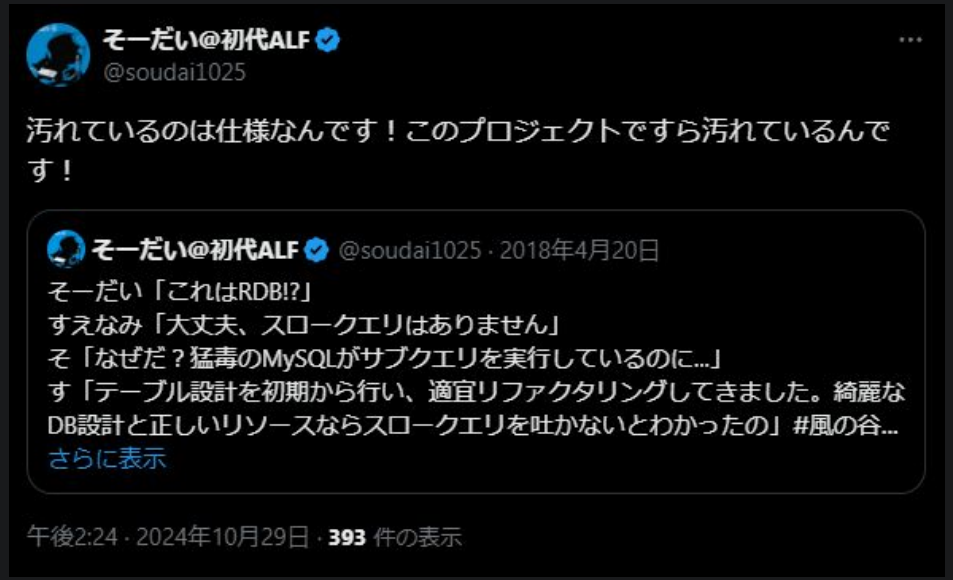
複雑な仕様がスロークエリや

ロングトランザクションを生む

複雑な仕様がRDBを殺す



複雑な仕様がRDBを殺す



そーだい@初代ALF
@soudai1025

汚れているのは仕様なんです！このプロジェクトですら汚れているんです！

> そーだい@初代ALF @soudai1025 · 2018年4月20日
> そーだい「これはRDB!?」
> すえなみ「大丈夫、スロークエリはありません」
> そ「なぜだ？猛毒のMySQLがサブクエリを実行しているのに…」
> す「テーブル設計を初期から行い、適宜リファクタリングしてきました。綺麗なDB設計と正しいリソースならスロークエリを吐かないとわかったの」#風の谷…
> さらに表示

午後2:24 · 2024年10月29日 · **393** 件の表示

複雑な仕様がRDBを殺す

これは本当に闇が深い

複雑な仕様がRDBを殺す

ECサイトの例

複雑な仕様がRDBを殺す

ECサイトの例
↓
購入のボタンがめちゃ重い

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

これを全部
1つのトランザクションまとめる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

いきなりUPDATEとかで
ロックを取ると後続が詰まる

これを全部
1つのトランザクションまとめる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーン
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

処理の数だけクエリを投げて集計する

これを全部
1つのトランザクションまとめる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーン
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

処理の数だけクエリを投げて集計する

これを全部

まとめる

無理に1回のクエリで対応しようとして、
超複雑なクエリが生まれる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーン
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

処理の数だけクエリを投げて集計する

これを全部

まとめる

無理に1回のクエリで対応しようとして、
超複雑なクエリが生まれる

さらにN+1のループがあったりする

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手
7. ポイン
8. 購入履歴の追加
9. 通知処理

外部APIだったりする

これを全部
1つのトランザクションまとめる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手
7. ポイン
8. 購入履歴の追加
9. 通知処理

これを全部
1つのトランザクションまとめる

外部APIだったりする

しかも時々めちゃレスポンスが遅い
ISUCON9の予選問題がこれ

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント
8. 購入履歴
9. 通知処理

これを全部
1つのトランザクションまとめる

ここも外部APIだったりする
失敗したりするとロールバックでやり直し
ここが終わってやっと画面遷移が終わる

複雑な仕様がRDBを殺す

全部を同時実行する必要はない

複雑な仕様がRDBを殺す

全部を同時実行する必要はない

↓

ユーザが知りたいのは購入できた事実

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

在庫を確保できば後続は非同期にできる

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

後続で処理して
失敗したら購入を取り消す

複雑な仕様がRDBを殺す

1. 商品の在庫を確保
2. ポイントの利用
3. クーポンの利用
4. キャンペーンの確認
5. 決済処理
6. 配送手続き処理
7. ポイント付与
8. 購入履歴の追加
9. 通知処理

商品次第ではここさえ確定できれば、
あとは非同期にできる

複雑な仕様がRDBを殺す

ロングトランザクション対策

複雑な仕様がRDBを殺す

ロングトランザクション対策
↓
分割する

複雑な仕様がRDBを殺す

？？？「でも全部同じテーブルなんですよ」

複雑な仕様がRDBを殺す

複雑な仕様がRDBを殺す

？？？「でも全部同じテーブルなんですよ」
↓
ライフサイクルに合わせてテーブルは分割

複雑な仕様がRDBを殺す

ちゃんと正規化する

責務を正しく分ける

ここまでシンプルな実装を目指しましょうと強調してきましたが、「シンプルな実装」とはなんでしょうか。RDBMSを使う上でシンプルな実装のヒントは正規化です。正規化のコツは次のように表現できます。

- 事実だけを保存する
- 重複がない
- 不整合がない
- nullがない

これらを意識して設計していくとシンプルな設計に近づいていきます。

また正規化を行う際はここまで説明したとおり、種別と状態を考えることも重要です。ライフサイクルが違うデータは往々にして状態や種別が異なります。場合によってはnullになるようなカラムやUPDATEが必要なレコードは状態を持っている可能性があります。こうしたテーブルが見つかった場合はより深く考察する必要があります。

アジャイル開発とデータベース設計 - 変化に対応するシンプルな実装のために必要なこと

— 2022-07-28

はじめまして。そーだい（@soudai1025）です。私は普段は技術コンサルティングや受託開発を請け負う合同会社HaveFunTechの代表として、また、予防治療の自社サービスを展開する株式会社リンケージのCTOという二足の草鞋を履き、日々、さまざまなWebサービスの開発に携わっています。

これまでの開発経験のなかで、データベース設計に関わるさまざまな問題に遭遇してきましたが、本稿ではとくに、アジャイル開発時に発生しやすい問題とその対処についてお伝えしたいと思います。開発の現場で目にしやすい実装におけるアンチパターンを示しつつ、アジャイルという指針を維持しながら、対処となるデータベース設計についてご紹介します。

https://agilejourney.uzabase.com/entry/2022/07/28/103000

そして最後にINDEXの数にも注目しましょう。主キーは必ずありますが、外部キー制約とユニーク制約を除いたINDEXは主に検索のために必要なINDEXです。検索のWHEREの対象の数だけそのテーブルの責務が大きいといえ、4つ以上のINDEXが必要な場合も同じく深く考察する必要があります。隠れた状態をWHEREで絞り込んでいたり、種別をWHEREで絞り込んでいるケースが見えてくることがあります。

このようにシンプルな設計を目指して考察を繰り返していくことが重要です。そして同じくらい重要なこととして認識すべきはイージーとシンプルは両立できる、ということです。 シンプルを目指し考察を繰り返すことがまさにデータモデリングであり、変化に強い設計につながっていくのです。

アジャイル開発とデータベース設計 - 変化に対応するシンプルな実装のために必要なこと

— 2022-07-28

はじめまして。そーだい（@soudai1025）です。私は普段は技術コンサルティングや受託開発を請け負う合同会社HaveFunTechの代表として、また、予防治療の自社サービスを展開する株式会社リンケージのCTOという二足の草鞋を履き、日々、さまざまなWebサービスの開発に携わっています。

これまでの開発経験のなかで、データベース設計に関わるさまざまな問題に遭遇してきましたが、本稿ではとくに、アジャイル開発時に発生しやすい問題とその対処についてお伝えしたいと思います。開発の現場で目にしやすい実装におけるアンチパターンを示しつつ、アジャイルという指針を維持しながら、対処となるデータベース設計についてご紹介します。

https://agilejourney.uzabase.com/entry/2022/07/28/103000

複雑な仕様がRDBを殺す

あじぇんだ

1. 自己紹介
2. 複雑な仕様がRDBを殺す
3. 特効薬：仕様を削る
4. まとめ

特効薬：仕様を削る


ポストする
そーだい@初代ALF
@soudai1025
プロモーションする
パフォーマンスチューニング、仕様を減らすのが一番コスパいいんだよなあ。
午後1:49 · 2020年10月5日
ポストのエンゲージメントを表示
15
73

特効薬：仕様を削る

ずっとこれ言ってる

特効薬：仕様を削る

ずっとこれ言ってる
↓
代表格を紹介します

特効薬：仕様を削る

ページャの原罪

特効薬：仕様を削る

ページャの原罪
↓
ページャの業は深い

特効薬：仕様を削る

1つ目

対象総件数の表示

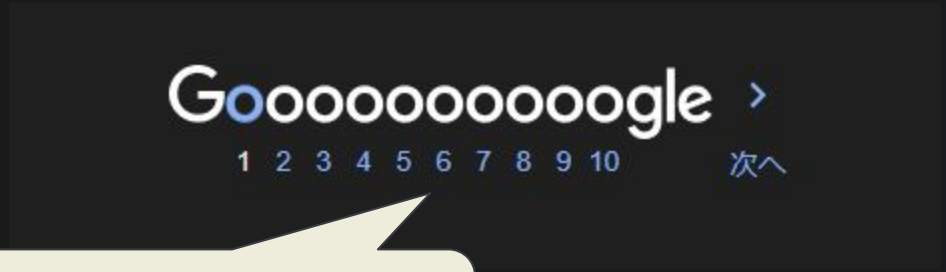
特効薬：仕様を削る


Google™
Search 1,326,920,000 web pages
曽根壮大
Advanced Search
Preferences
Google Search
I'm Feeling Lucky
Google Web Directory
the web organized by topic

特効薬：仕様を削る


Google
Search 1,326,920,000 web pages
曽根壮大
Advanced Search
Preferences
このcount(*)が不要なクエリを生む

特効薬：仕様を削る



今はGoogleも表示していない

特効薬：仕様を削る

sql_culc_found_rows の呪い

特効薬：仕様を削る

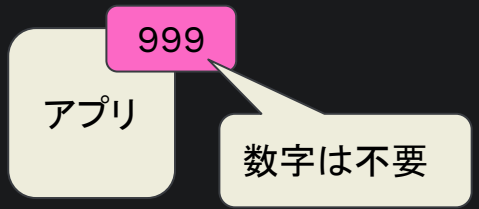
sql_culc_found_rows の呪い

MySQLで全体件数を取得するために使われていた
MySQL 8.0.17で非推奨になった

特効薬：仕様を削る

通知の未読件数とかも一緒

基本的には未読かどうかわかれば良い



特効薬：仕様を削る

INDEXが活用できないcount()は

めちゃめちゃ遅い

テーブルスキャンだから

特効薬：仕様を削る

INDEXが活用できないcount()は

めちゃめちゃ遅い

テーブルスキャンだから

複雑な検索フォームの場合
テーブルスキャンになるケースで刺さる

特効薬：仕様を削る

2つ目

最終ページのリンク

特効薬：仕様を削る


ポストする
kawasima@99卒
@kawasima
「ページネーションの"最後のページへ" のリンクは無くしましょう」は鉄板
1
2
3
4
5
...
242
次へ
そーだい@初代ALF
@soudai1025 · 10月15日
パフォーマンスチューニングには仕様変更が一番効くって話をしようと思う。
x.com/Keisuke69/stat...
午後6:17 · 2024年10月15日 · 114.3万 件の表示
12
430
2,619
885

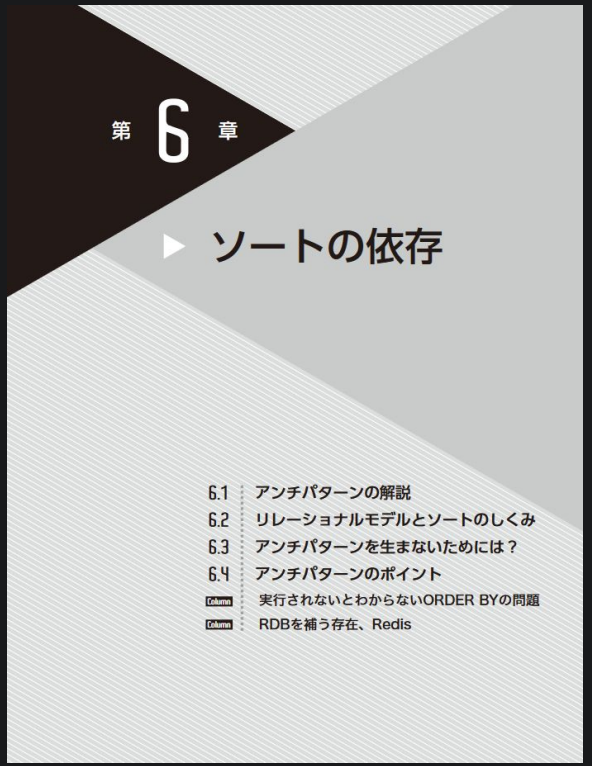
特効薬：仕様を削る

ORDER BYの

LIMIT OFFSETに刺さる

特効薬：仕様を削る


Software Design plus
失敗から学ぶ
RDBの
正しい歩き方
曽根 壮大
RDB The Right Way
データベースの迷宮
失われた事実
やり過ぎたJOIN
効かないINDEX
フラグの闇
ソートの依存
隠された状態
JSONの甘い罠
強過ぎる制約
転んだ後のバックアップ
見られないエラーログ
監視されないデータベース
知らないロック
ロックの功罪
簡単過ぎる不整合
キャッシュ中毒
複雑なクエリ
ノーチェンジ・コンフィグ
塩漬けのバージョン
フレームワーク依存症
MySQL・PostgreSQLの
設計と運用を見直す
技術評論社


第 6 章
ソートの依存
6.1 アンチパターンの解説
6.2 リレーショナルモデルとソートのしくみ
6.3 アンチパターンを生まないためには？
6.4 アンチパターンのポイント
Column 実行されないとわからないORDER BYの問題
Column RDBを補う存在、Redis

特効薬：仕様を削る

```
LIMIT 10
```

のとおり、10件しか取り出しません。
　ですがrowsの項目に注目してください。そうです！　510件も取り出しています。これは、

```
LIMIT 10 OFFSET 500
```

のため、500行目まで順番に確認して、さらにそこから10件を取り出したため、全体としては500+10＝510件のデータを取り出しているのです。つまり、

```
LIMIT 10 OFFSET 100000
```

なら100,010件を取り出すことになります。これが今回のアンチパターンの大きな原因になっています。

特効薬：仕様を削る

最終ページに行きたい？

昇順、降順で良い

特効薬：仕様を削る

そもそもユーザが必要なのは

検索フォームだったりする

特効薬：仕様を削る

そもそもユーザが必要なのは

検索フォームだったりする

これもこれで闇が深いが …
部分一致とかORとか…

特効薬：仕様を削る

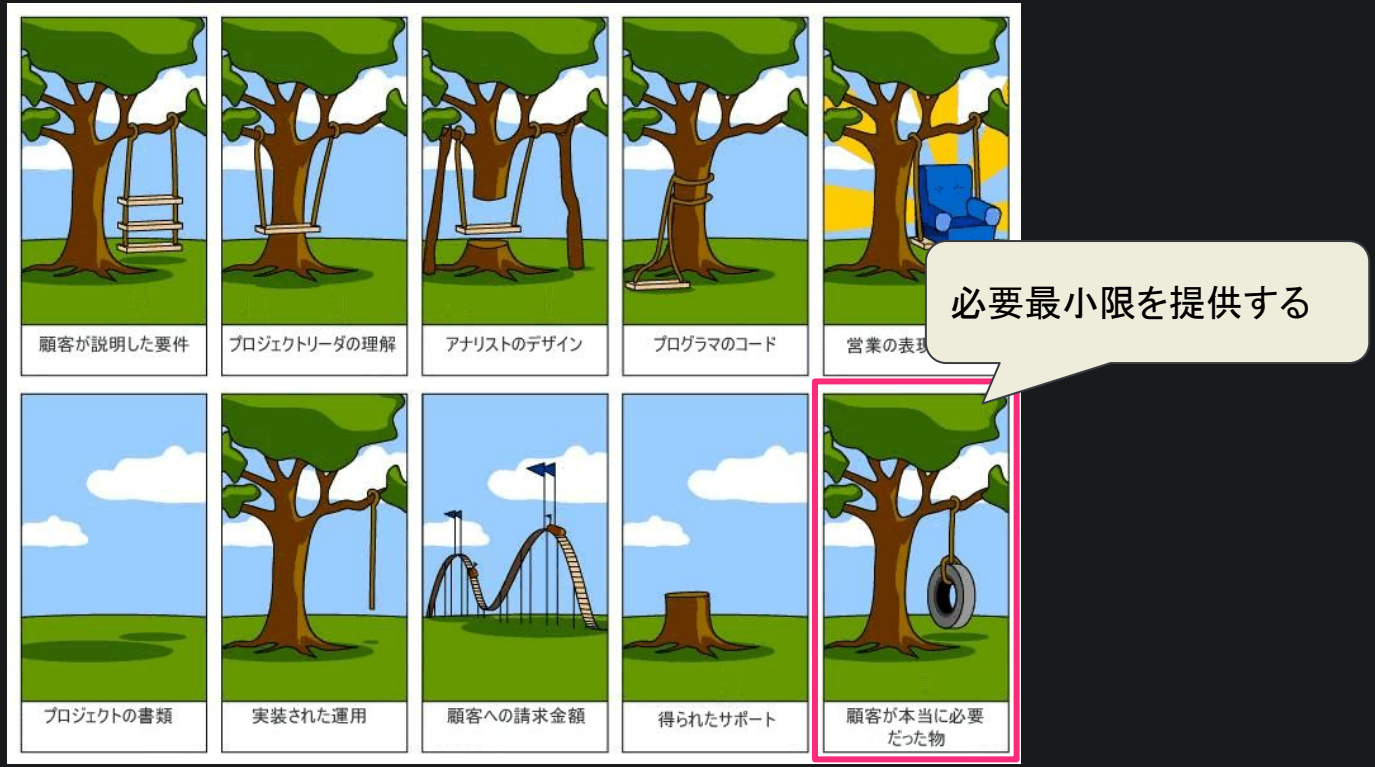
これらは単純に仕様を消せば

実装不要でパフォーマンスも改善する

特効薬：仕様を削る


ポストする
そーだい@初代ALF
@soudai1025
プロモーションする
パフォーマンスチューニング、仕様を減らすのが一番コスパいいんだよなあ。
午後1:49 · 2020年10月5日
ポストのエンゲージメントを表示
15
73

What is it?


顧客が説明した要件
プロジェクトリーダの理解
アナリストのデザイン
プログラマのコード
営業の表現
必要最小限を提供する
プロジェクトの書類
実装された運用
顧客への請求金額
得られたサポート
顧客が本当に必要だった物

特効薬：仕様を削る

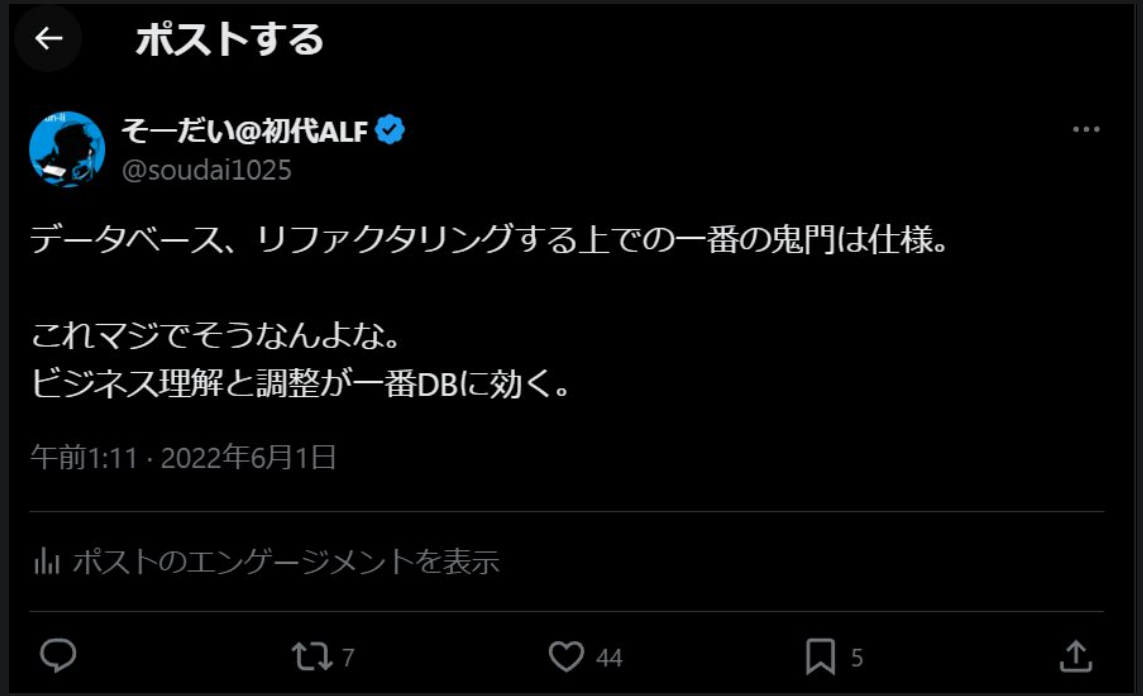
本当に必要なモノだけを

実装していますか？

あじぇんだ

1. 自己紹介
2. 複雑な仕様がRDBを殺す
3. 特効薬：仕様を削る
4. まとめ

複雑な仕様がRDBを殺す



まとめ

Simple is Beautiful

まとめ

Simple is Beautiful

↓

常に追求した者だけが辿り着ける

まとめ

本当に必要なモノを実装する

まとめ

本当に必要なモノを実装する
↓
そのためにプロダクト、ドメインと向き合う

まとめ


ポストする
そーだい@初代ALF
@soudai1025
仕様を満たすコードを書くのではなく、要求を満たすコードを書くことが大事だし、要件に合わせて設計するのではなく、要求に合わせて要件を整理するのが大事なんだよな。 #phpconfuk
午後4:16 · 2023年6月24日 · 1.3万 件の表示
ポストのエンゲージメントを表示
16
56
8

まとめ

質を追求した結果

パフォーマンスは後からついてくる

まとめ

良いソフトウェアを作ろう

目の前のプロダクトに向き合おう

まとめ

昨日の自分に誇れる

今日の自分になろう

まとめ

ご清聴ありがとうございました