

負債と言わないことが 負債と向き合うこと

高橋健一 @kenchan / GMOペパボ株式会社
2023/11/21 技術的負債に向き合う Online Conference

GMOペパボ

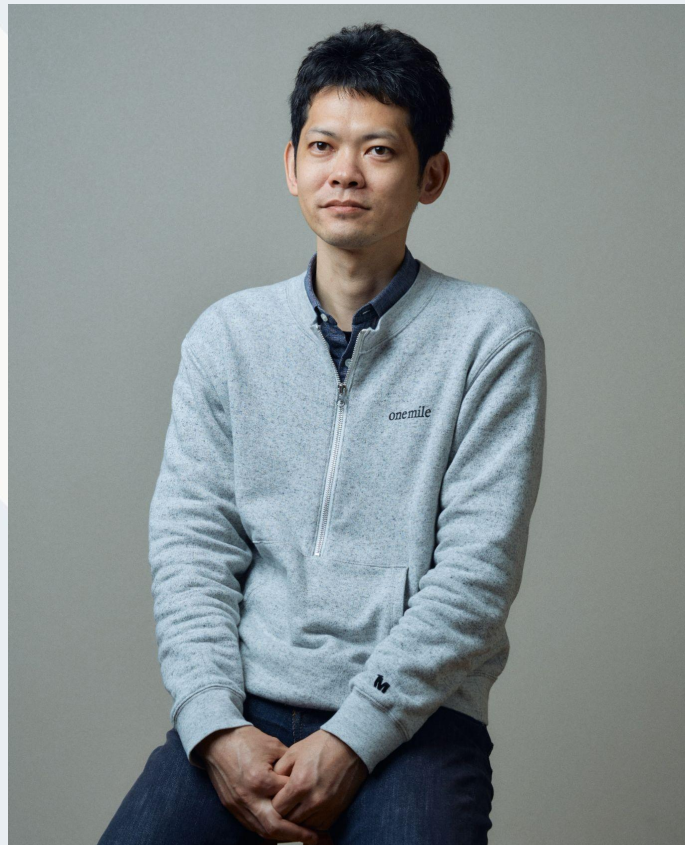
GMOペパボ株式会社 技術責任者
兼 EC事業部シニアエンジニアリングリード

高橋 健一 Kenichi Takahashi

Rubyとアジャイルなソフトウェア開発が好きな
ソフトウェアエンジニア。

日々の仕事は、EMとして全社のエンジニア組織
作りと、カラーミーショップを運営するEC事業
部における実質CTO。

インターネットでは @kenchan で活動中。フォ
ロワー、チャンネル登録、高評価よろしくおねが
いします。



1. ”負債”のメタファーを乗り越える
2. 15年以上続くサービスで取り組んでいること
3. AIが前提となる時代における技術的”資本”の重要性

“負債”のメタファーは金融システム開発の中で生まれた

- Ward Cunninghamは、問題領域へのチームの理解と目の前のシステムのギャップを負債と呼んだ。
- 問題領域への理解はかならずチームの理解が先行する。それをシステムに反映するにあたり必ず時差が生じる。
- 負債というメタファーは、金融システムの責任者が理解しやすい言葉で状況を説明するために考案した。

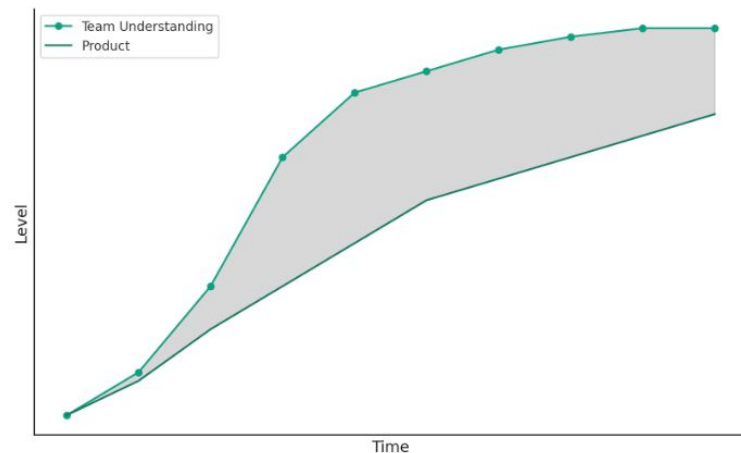
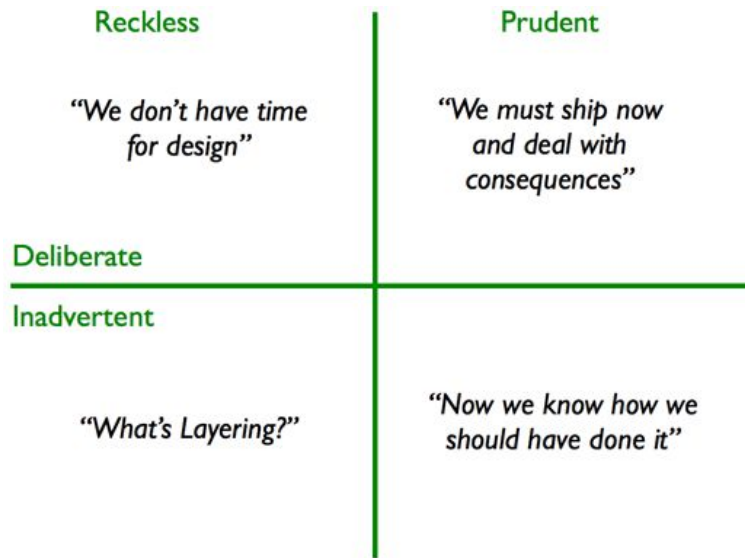


図. 時間の経過によってギャップが生まれるイメージ

(技術的)負債のメタファーの乱用、分類、再考が進む

- 負債というメタファーが優れていたが故に議論が白熱する。
 - 例:稚拙なコードは負債か？
- Martin Fowlerの四象限の整理やSoftware Engineering Instituteによる12の分類など整理が進む。
- メタファーを利用しつつ、より具体的な問題に対して向き合うことが重要となった。



現代では負債を借り入れずにシステム開発をすることは不可能(1)

- チームが問題領域を完全に理解し、それを完璧にシステムに反映できたら……！？
 - “なるほど完璧な作戦っすね——っ 不可能だという点に目をつぶればよお～”
- 市場への投入が遅れば遅れるほどその価値は目減りし、いつか損益分岐点を下回る。
 - 例: 競合に先を越される。市場がなくなる。

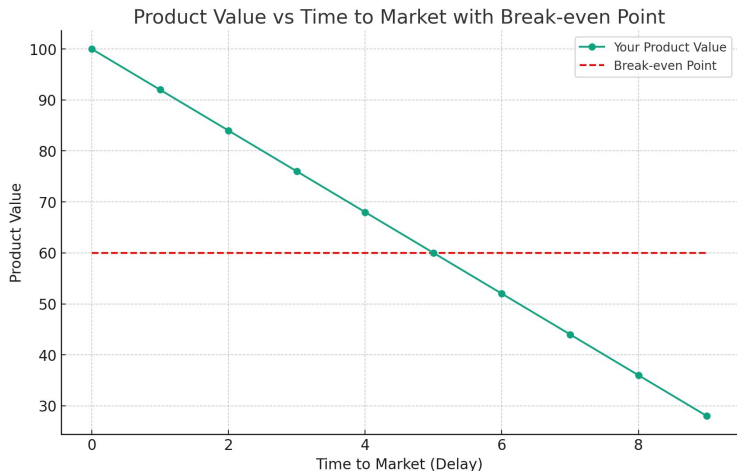


図. 市場価値の損益分岐点のイメージ

現代では負債を借り入れずにシステム開発をすることは不可能(2)

- ソフトウェアは何もしないと壊れる。
- ビジネス的にも「何もしない」と顧客からは「壊れて見える」ようになる。
- ある瞬間において完璧なシステムがあったとしても、世の中の変化によってギャップが生まれてしまう。

ソフトウェアは何もしないと壊れる

GMOペパボ

- 「何もしてないのに壊れた...」
 - ソフトウェアを動かす環境は絶えず変化している
 - OS、ミドルウェア、ハードウェア
 - 脆弱性、サポートコストなど環境の変化からは逃れられない
- 現代のソフトウェアは複雑怪奇な組み合わせの中でなんとか動いている
 - 「何もしていないから壊れる」「それはそう」

負債のメタファーは銀の弾丸ではない

- 経営者や事業責任者にとっての負債と”技術的負債”で伝えたいことのギャップ。
 - 負債(他人資本)は事業をレバレッジさせるための重要な道具であり、具体的な金額や返済期限として認識可能。
 - “技術的負債”で伝えたいことは、目に見えないものであることが多い。
- メタファーは共通の理解を手助けするが、課題を解決してくれるわけではない。
 - 「技術的負債を返済しなければなりません」「なるほど。どんどんやろう！」
 - そんなことありますか？
- デリバリー速度を低下させる要因であるなら「(技術的)負債である」で留まらずに詳細化することが解決に直結する。

ここまでのまとめ

- 負債というメタファーは金融システム開発において責任者とのコミュニケーションのために生まれた。
- 「負債」という言葉によってかえってコミュニケーションコストがかかっている場合も散見される。
- 事業としてプロダクト開発をしている限り、負債を0にすることはできないし、それを目指すべきでもない。
- デリバリー速度が事業成長に大きく影響があることは周知の事実であり、具体的な課題について議論することが真に負債と向き合うことではないか。

1. ”負債”のメタファーを乗り越える
2. 15年以上続くサービスで取り組んでいること
3. AIが前提となる時代における技術的”資本”の重要性

カラーミーショップ ミッション

日本の商いを なめらかに。

カラーミーショップは、GMOペパボが17年以上運営している
利用者満足度No.1※のネットショップ作成サービスです。
ネットショップに必須の機能、まるごと搭載。
あらゆる業態に対応したカスタマイズ性で、ビジネスの成長を支えます。

ネットショップ作成サービス **No.1**※1
利用者満足

売上げが期待できる **No.1**※2
ネットショップ作成サービス

※インターネットアンケートによりGoogle検索で検索上位に表示されるネットショップ作成サービス9社について比較。日本マーケティングリサーチ機構調べ。2022年1月調査。



18年目を迎えるECサイト構築サービス「カラーミーショップ」

- ロリポップ！のECカート機能を切り出して2005年にリリース。
- バージョン管理がない時代からメンテナンスされているコードベース
 - 最古のコミットは2010年7月2日、Subversionへの「新規作成」というコミット
- 自社データセンターとパブリッククラウドを併用したハイブリッドクラウド環境
- バックエンドはPHPとRuby、フロントエンドはVue.js、Angular、jQuery
- “技術的負債”と日々向き合いプロダクト開発を進めている。

技術的負債を返済するための3+1つのアプローチを愚直に実行する

- リファクタリング
 - ボトムアップにコードを綺麗にしていく
- リアーキテクティング
 - 大きなシステムをサブシステムに分解し、サブシステム単位で置き換えていく
- リプレイス/リライト
 - ゼロから書き直す
- 葬り
 - ソースコードを削除して機能を無くす

テストを文化とするための研修とその実践(リファクタリング)

2022-07-22

t_wadaさんによるTDDワークショップを開催しました

TDD programming software 研修

まとめ: TDDのスキル

- 問題を小さく分割する
- 歩幅を調整する
 - テスト → 仮実装 → 三角測量 → 実装
 - テスト → 仮実装 → 実装
 - テスト → 明白な実装
- テストの構造化とリファクタリング

ご清聴ありがとうございました

プランアップ料金が正しく計算されていることを確認するテストを追加

← Merged ayami merged 7 commits into master from add_test_planup on Oct 10

Conversation 5 Commits 7 Checks 19 Files changed 5



ayami commented on Sep 20 • edited by ryoma123

何を解決するのか

colorme/elephant#52885 の調査用
毎月1日は月額料金をまるっと買うため、その計算が正しく行われていることを確認するテストコードを追加した。
(申込み部分はラージプランアップのみ追加)

リリース時期

レビューが通り次第

レビューを依頼する前に

- PHP ファイルの変更がコーディングルールに従っていることを確認した
- HTML テンプレート、Javascript ファイルの変更がコーディングルールに従っていることを確認した
- SCSS、CSS、Javascriptファイルの変更時はHTML側の読み込みのクエリストリングを修正する

HTMLやPHPを変更した際は下記の部分についても確認してください

対応方法についてはwikiをご参照ください

Reviewers

ryoma123

Assignees

No one—assign you

Labels

None yet

Projects

None yet

Milestone

No milestone

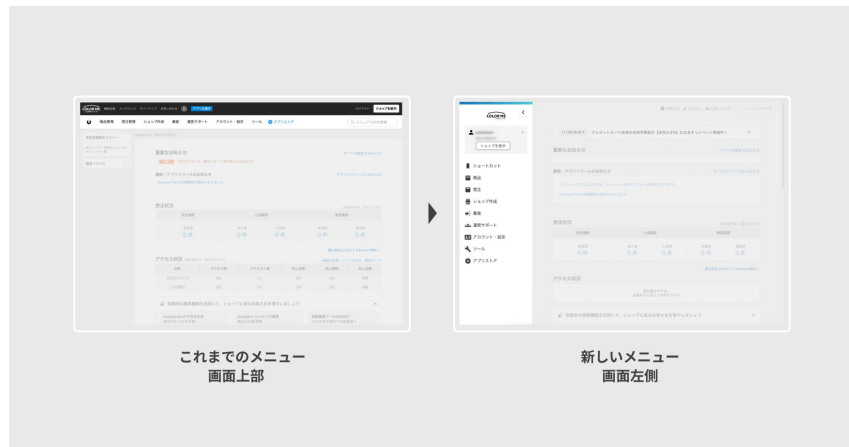
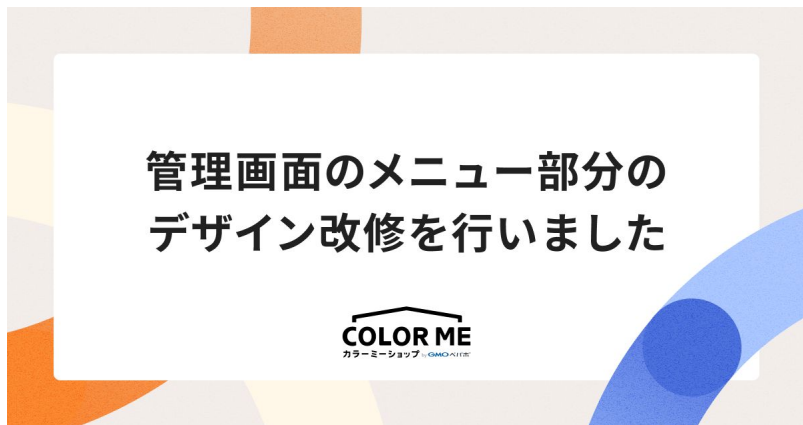
Development

Successfully merge issues.

None yet

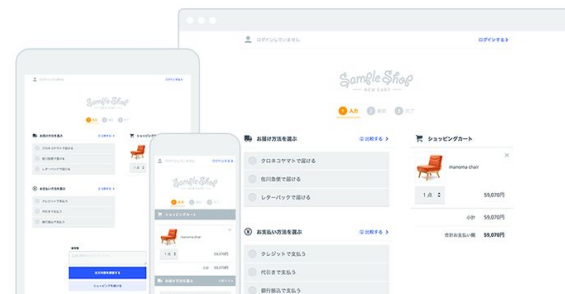
画面をリニューアルする(リアーキテクティング)

- PHPとjQueryによるMPAにVue.jsを段階的に導入



ショッピングカートを作り変える(リプレース+葬り)

- システムとして一定の独立性がある場合はリプレースも検討する。
- 避けられるなら避けたほうがよいが、やるのであればプロモーション等まで最大活用する。
- 葬る場合はユーザメリットを提示して移行を促す。
 - 例: より便利な決済方法を提供する



ユビキタス言語を作る(ドキュメンテーション)

- ドキュメントも問題領域への理解を反映させることができる対象。
- コードやアーキテクチャの変更が困難な場合でもas-isとto-beを残す。

ユビキタス言語

Type: **ドメインモデル** **ドキュメンテーション**

ステータス: **更新中**

担当者: Emery

最終更新日: 2023/06/01 12:13 PM

Add a property

Add a comment...

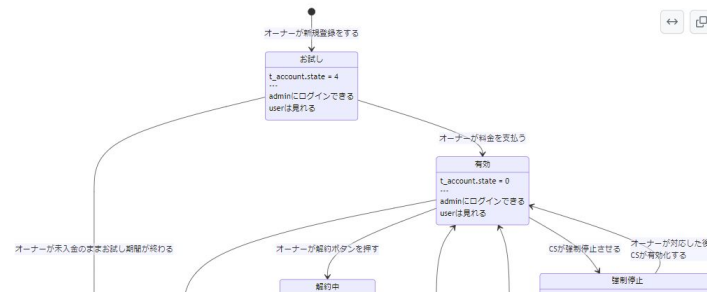
7 bookmarks

何か変更したい点や気づきがありましたら、その名前のページコメントに記載してください。担当の人と話し合えたら、担当の人が変更修正をさせていただきます。

As is	現在	To be	将来
売上	Sales	ドメインオブジェクト	Observer型にしたもの。プラットフォームと、機能別のプラットフォームでふしな表現方法を用いて、表現される。
ショップインフォメーション	Shop information	ドメインオブジェクト	ネットワーク上に存在する情報(単一概念)。元々、店舗の属性を指す。

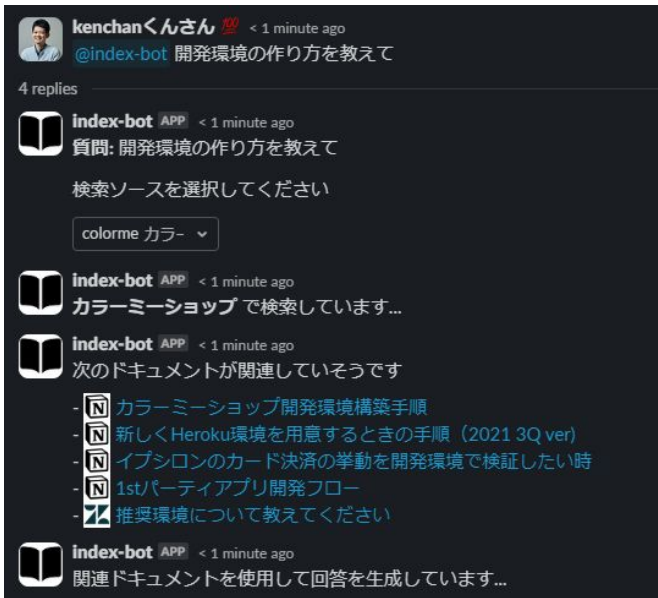
アカウントの状態遷移図

以下アカウントの状態遷移図を示します、その状態のときの区別の仕方や、なにが出来るかも書いてあります。



ドキュメントを腐らせない唯一の方法は使うこと

- ドキュメントは腐りやすい。
- ドキュメントの共同所有は人類の夢。
- 夢に近づくためには、使うことと問題解決した経験を積み重ねること。
- ドキュメントを自然言語で検索できる slackbot を導入し運用している。



負債に向き合う文化づくり ～共通認識を育む～

- 負債というメタファーが、ドメインスペシャリストとのコミュニケーションのために生まれたものであれば、その考え方自体を応用できるはず。
- 会社、部門、プロジェクトにおいて重要視されていることはなにか。
 - ユーザ体験、デリバリーの速度、セキュリティ、コストetc
- 目の前にある負債は、何にどのような影響があるのかを説明できるか。

- 共通の認識を育むきっかけにできるものを探す。

文化を育むためには既にある文化でレバレッジをかける

- 負債のメタファーのままでは、借りてきた言葉の域を出ない。目の前の課題と自分達の文化を混ぜ合わせてやるべきことやる。
- GMOペパボでは「やっていき・のっていき」「ナイストライ」という文化と、その前提としての「ユーザファーストであること」を活用する。
 - やっていき≡リーダーシップ
 - のっていき≡フォロワーシップ
 - ナイストライ≡挑戦を自体を賞賛する

ここまでのまとめ

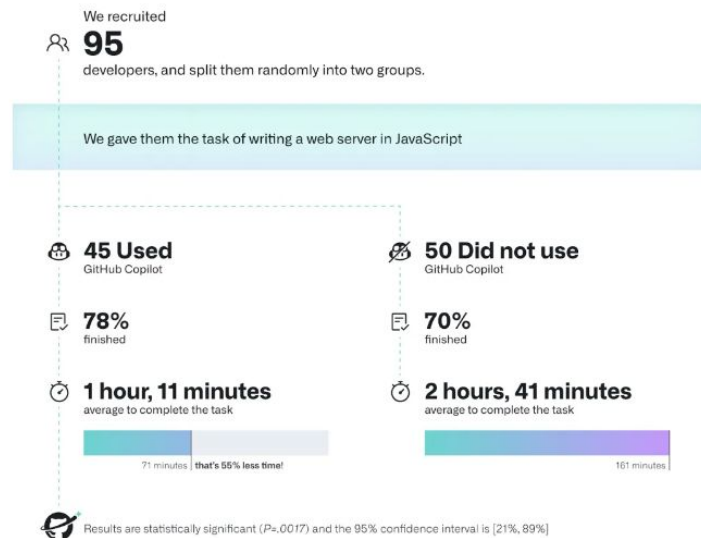
- 18年目を迎えるカラーミーショップは多分に漏れず様々な技術的負債があり、それに対して日々改善に取り組んでいる。
- 取り組みは至って普通で、リファクタリング、リアーキテクティング、リプレース、葬りを組み合わせている。
- 負債というメタファーを活用しつつ、課題を解決するためには自分達の内にある大切にしていることとの繋りを活用する。

1. ”負債”のメタファーとその限界
2. 15年以上続くサービスで取り組んでいること
3. AIが前提となる時代における技術的”資本”の重要性

GitHub Copilot利用によって開発速度が55%向上

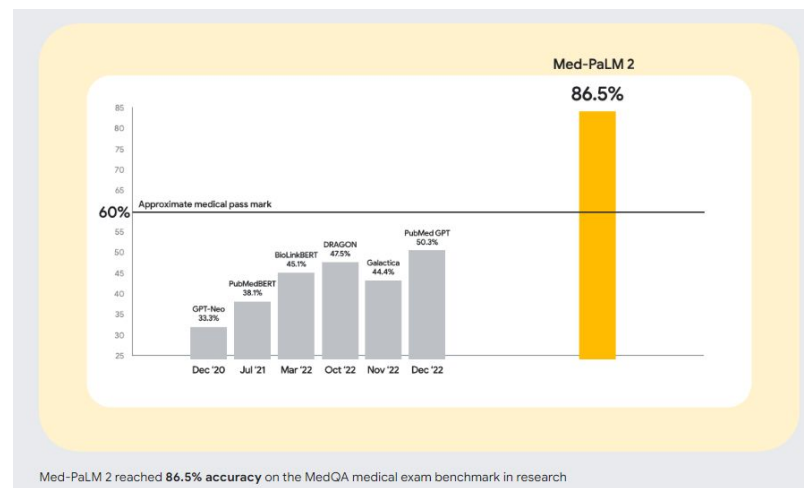
- HTTPサーバのコーディングにおいてはGitHub Copilotの有無で開発速度に大きな差があった。
- 一方で、私達が日々向き合っている課題はHTTPサーバの実装よりもはるかに不確実性が高い上、事業ドメインの深い知識が必要。

図：実験のプロセスと結果の概要



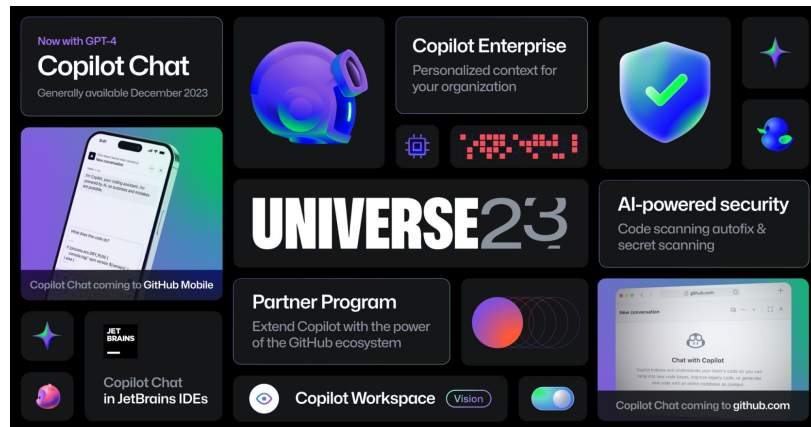
ドメイン特化LLMやドメイン固有の知識を与える手法で精度が上がる

- Med-PaLM 2は医療ドメインに特化したLLMであり、米国医師免許試験の問題で「エキスパート」レベルのパフォーマンスを達成している。
- LangChainやLlamaIndex、ChatGPTのCode interpreterなど外部から固有の知識を与えることで精度が向上する。



事業ドメインとプロダクトを理解したAIと一緒に開発する

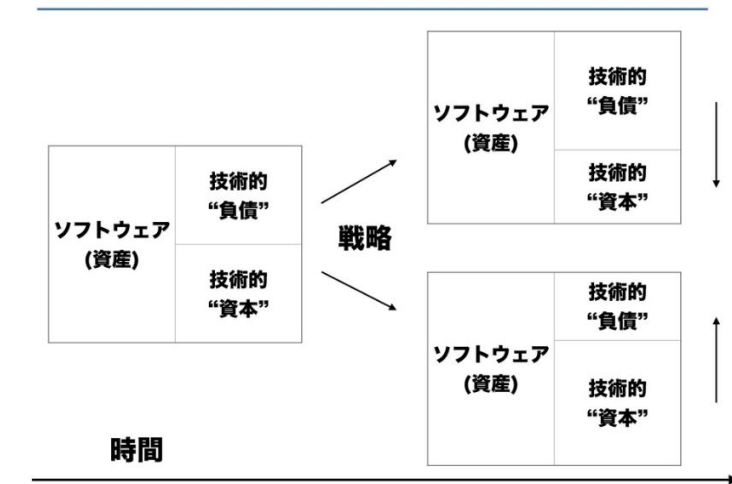
- 社内の資産を活用することで生成系AIの価値はさらに大きなものになる。
 - ユビキタス言語、データベース構造、ナレッジベースを活用できるか。
- ソフトウェアの資産価値が競争力に直結する時代に。



「技術的負債を返済する」から「技術的資本の割合を増やす」へ

- ソフトウェア資産の構成要素は負債と資本。
- チームの理解とプロダクトのギャップが負債であるなら、正しく反映されたものが資本。
 - 良いコード
 - 良いアーキテクチャ
 - 良いドキュメントetc
- 負債を返却するというマインドセットから資本を増やすマインドセットへ転換。

資本構成戦略



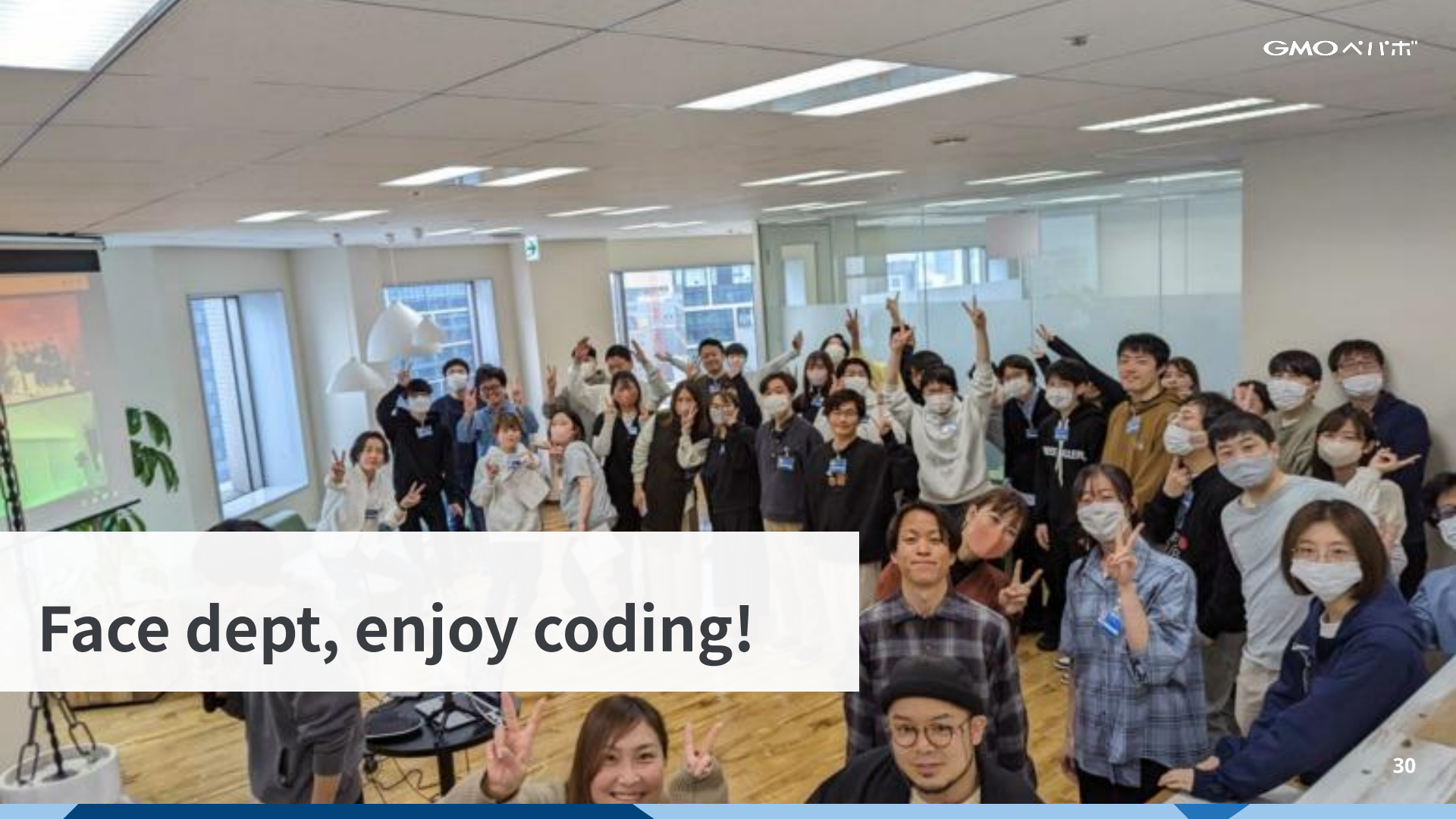
負債を減らす or 資本を増やす

- 具体的にやることは一緒なので言葉遊びであるかもしれない。
- 一方で、言葉を変え、マインドセットを変えることで変化があるかもしれない。
- 生成系AIの登場で(技術的)資本は組織の生産性に直結することは確実。
- 良いコード・アーキテクチャ・ドキュメントの価値はスーパーインフレへ。

ここまでのまとめ

- 生成系AIの活用は企業の競争力の源泉となる可能性が高い。
- プロダクト開発においてもコード生成をはじめ一定の効果が報告されている。
- ソフトウェアの資産価値で生成系AIの効果にレバレッジをかける。
- 負債を返却するというマインドセットでうまくいなければ、資本を増やすというマインドセットへの転換はどうか

1. ”負債”のメタファーとその限界
2. 15年以上続くサービスで取り組んでいること
3. AIが前提となる時代における技術的”資本”の重要性



Face dept, enjoy coding!