杜甫々がプロジェクト管理で実践していた小ネタいろいろ

2025年11月29日

「とほほのWWW入門」

杜甫々（とほほ）

自己紹介

- 「とほほのWWW入門」管理人
  - 1996年9月からなので、もう30年目
- ハンドル：杜甫々（とほほ）
  - 本名は違います
- 広島生まれ。広島在住。カープファン
  - 今年の観戦成績は3勝2敗。
- インターネット歴
  - 1988年からなので38年
- インターネット老人会会長をめざしてます(ウソ)

「とほほのWWW入門」

https://www.tohoho-web.com/

- 1996年9月開設
- HTML/JavaScript/CSSなど
  Web関連技術紹介サイト

「とほほの○○入門」

Web関連技術を中心に紹介しています

「とほほの○○入門」

- 下記などの入門記事を紹介

プログラミング言語系
HTML入門（1996年）
JavaScript入門（1996年）
CSS入門（1997年）
Perl入門（2007年）
Java入門（2004年）
PHP入門（2013年）
Ruby入門（2014年）
Python入門（2014年）
Go言語入門（2020年）
C言語入門（2020年）
C#入門（2020年）
Rust入門（2020年）
Bash入門（2020年）
Haskell入門（2020年）
FORTRAN入門（2020年）
Scala入門（2021年）
AWK入門（2021年）
Kotlin入門（2021年）
Lisp入門（2022年）
8bitアセンブラ入門（2022年）

フレームワーク系
jQuery入門（2013年）
Bootstrap入門（2015年）
AngularJS入門（2015年）
Node.js入門（2016年）
Django入門（2018年）
Angular入門（2018年）
Vue.js入門（2018年）
React入門（2018年）
Laravel入門（2020年）
Ruby on Rails入門（2022年）
Next.js（2024年）

AI系
OpenAI（2025年）
Copilot in Win...（2025年）
NotebookLM（2025年）
Google Opal（2025年）
Google AI Studio（2025年）
Dify（2025年）
Codex（2025年）

その他系
文字コード入門（1996年）
色入門（1997年）
Cookie入門（1997年）
著作権入門（1998年）
HTTP入門（2005年）
フォント入門（2012年）
Docker入門（2016年）
セキュリティ入門（2018年）
暗号化入門（2021年）
正規表現入門（2021年）
DevTools（2024年）

どれがどの言語だったか覚えていられなくて、
コーディングする時は for文ひとつ書くにも
自分のリファレンスをにらめっこしてます...

略歴

- 1988年：大学を卒業してメーカー系ソフトウェア子会社に就職
- 1988年～：UNIX に TCP/IP を移植するプロジェクト
- 1990年～：ネットワーク管理システムの開発（35年プロジェクト）
- 1996年～：「とほほのWWW入門」開設（30年継続）
- 2000年～：性能管理システムの開発（短命）
- 2002年～：セキュリティ管理システムの開発（23年プロジェクト）
- 2013年～：クラウド管理システムの開発（12年プロジェクト）
- 2025年10月：退職

仕事の内容

- 1988年～：開発･コーディング
- 1996年～：PJリーダー
- 2003年～：管理職
- 2007年～：部長職
- 2008年～：技術畑に復帰 … 現在に至る

プロジェクト管理はPJリーダーにまかせて、
アーキテクト･技術支援など

プロジェクト管理･･･得意じゃないです

プロジェクト管理に関わる小ネタを紹介していきます。

- フォルダ構成
- 仕様書構成
- 開発ガイドライン
- その他

ちょっと古い話になるかも
しれませんが…

フォルダ構成

- 01_管理
- 02_資料
- 11_要件定義
- 12_概要設計
- 13_基本設計
  - XXX基本設計書_20251129.xlsx
- 14_詳細設計
- 20_開発･単体
- 31_内部結合テスト
- 32_外部結合テスト
- 33_総合テスト
- 40_リリース
- 50_保守
- 90_個人

← よくあるやつ

最初のバージョンは良い

バージョンアップ：v1.1, v1.2....

バグフィックス：v1.1.1, v1.1.2, ...

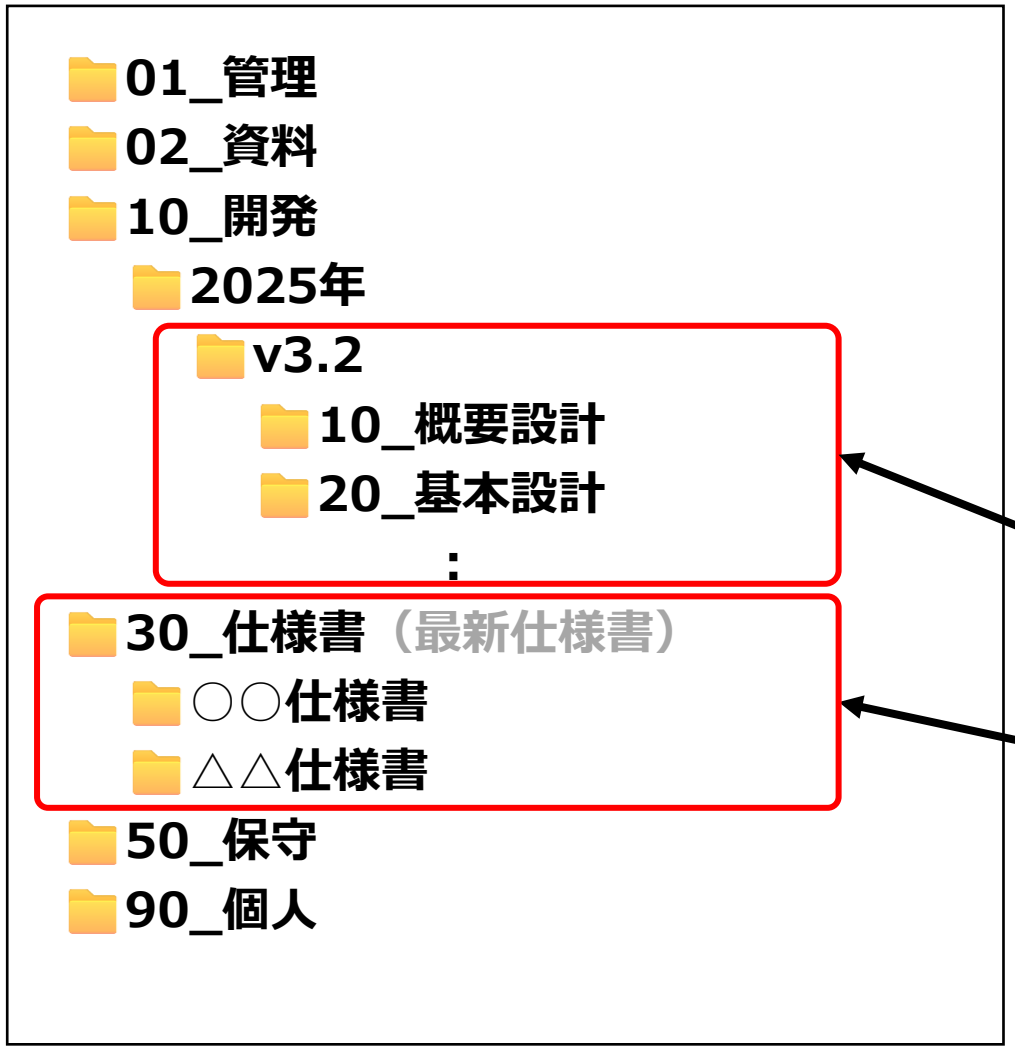
フォルダやファイル構成が **乱れ** てくる

10年フォルダ：10年経っても乱れないフォルダ構成

バージョンに依存するファイル群と
バージョンに依存しないファイル群を
完全に**分離**する

バージョンに依存するファイル群

最新仕様書はバージョンに依存しないファイル



最新仕様書はバージョンに依存しないファイルとして管理する

バージョン**依存**ファイル

バージョン**非依存**ファイル

バージョン依存/非依存両方の仕様書を書くのは面倒？

v3.2仕様書

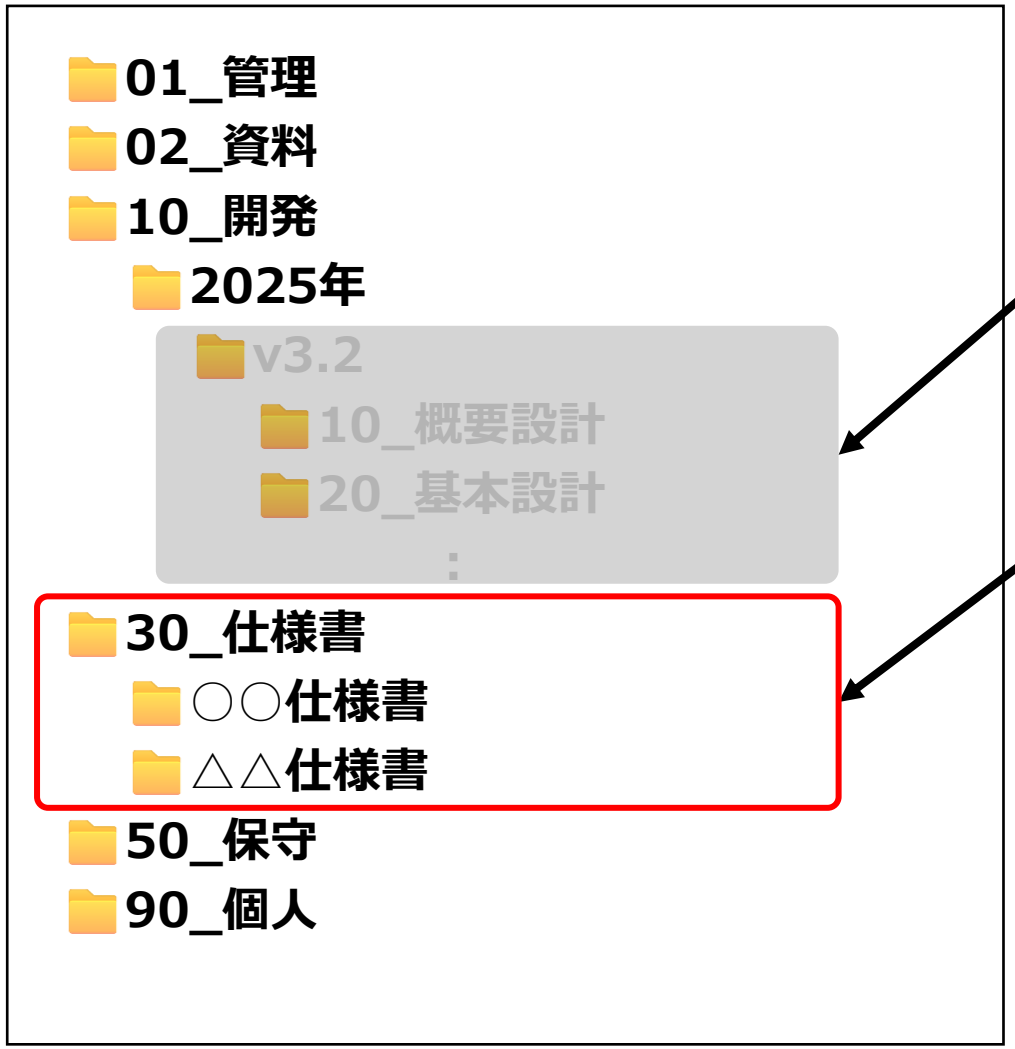
- [設定] 項目に [ZZ許可フラグ] を追加する
  → 詳細は 最新仕様書の…を参照

最新仕様書

[設定]

XX		
YY		
ZZ許可フラグ	…	…

リリースが終わったらバージョン依存仕様書は捨てる気持ちで



バージョン依存仕様書（ゴミ）

最新仕様書（大事）

10年後に保守しているメンバは
昔のバージョン依存仕様書なんて読まない

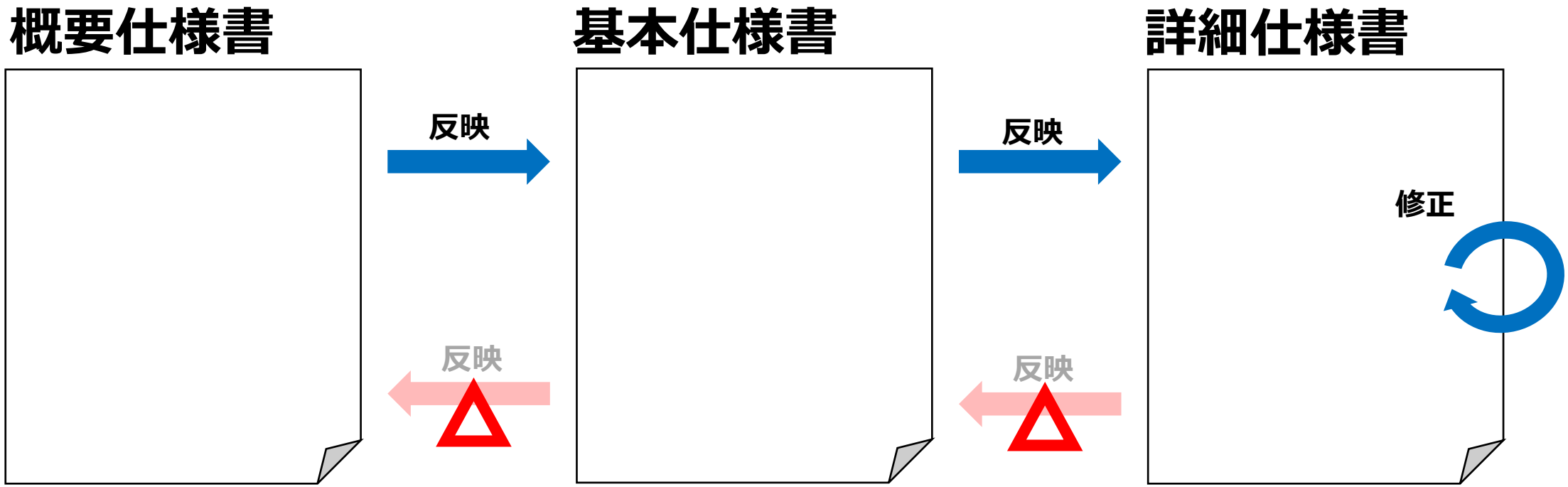
個人フォルダも捨てる気持ちで

個人フォルダ（ゴミ）

個人フォルダも年に一度捨ててしまう

- 個人フォルダ内に大事なデータを置かない
- 自分と次世代メンバくらいにしか伝わらない
- 大事なデータは正式な場所に置け

概要・基本・詳細仕様書



次工程における仕様変更は
前工程の仕様書に反映されないことがある
→ 仕様書の不整合

概要・基本・詳細仕様書は全部まとめて 仕様書 に

概要設計

基本設計

詳細設計

工程で分けずに
ひとつの仕様書にまとめちゃう
→ 不整合がなくなる

文書番号体系もシンプルにしました

XYZ-25-PO-DD-123

[プロジェクト]-[年度]-[モジュール]-[詳細設計]-[連番]

XYZ-S1234　：設計仕様書
XYZ-T1234　：テスト仕様書・報告書
XYZ-M1234　：手順書
XYZ-X1234　：その他

- 「詳細は S1234 を参照」などリンクが楽
- 検索しやすい
- 年度やモジュールは採番台帳で見ればよい
- RFC 方式ですね

仕様書、結構Excel使ったりしてます…


SPC-S0001

○○○システム仕様書

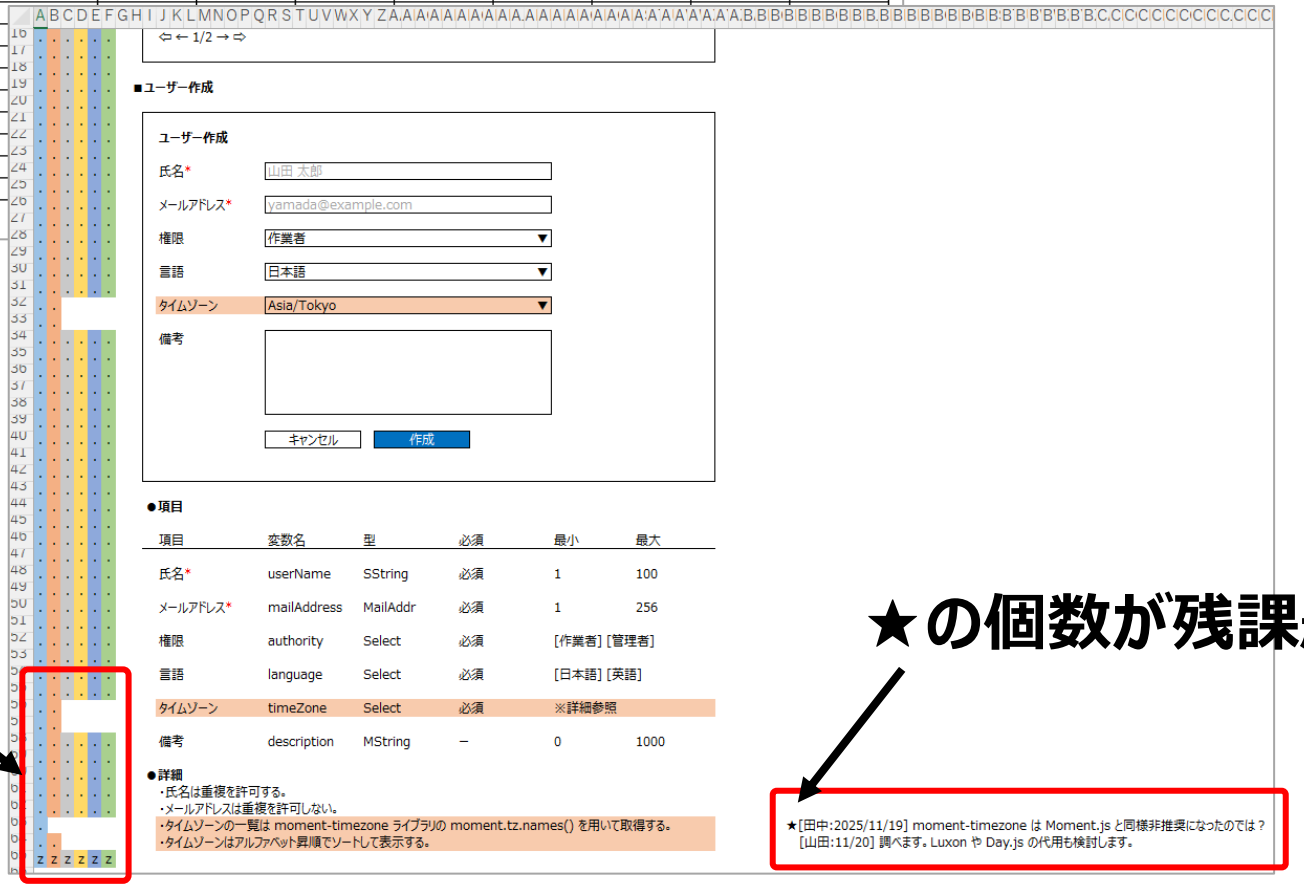
○○株式会社
○○事業部
○○プロジェクト

改版日	改版内容	承認	査閲	作成
2025/11/29	※改版履歴参照			山田

. … 作成完了
. … 査閲完了
. … 承認完了
. … 開発チーム確認完了
. … 実装完了
. … 評価完了


■集計

シート名	行数	A 作成	B 査閲	C 承認	D 開発	E 実装	F 評価	★ 課題
ダッシュボード	27	0	0	0	0	0	0	0
ユーザー管理	65	0	1	6	6	6	6	1


⇦ ← 1/2 → ⇨

■ユーザー作成

ユーザー作成
氏名* 山田 太郎
メールアドレス* yamada@example.com
権限 作業者
言語 日本語
タイムゾーン Asia/Tokyo
備考
キャンセル 作成

●項目

項目	変数名	型	必須	最小	最大
氏名*	userName	SString	必須	1	100
メールアドレス*	mailAddress	MailAddr	必須	1	256
権限	authority	Select	必須	[作業者] [管理者]	
言語	language	Select	必須	[日本語] [英語]	
タイムゾーン	timeZone	Select	必須	※詳細参照	
備考	description	MString	–	0	1000

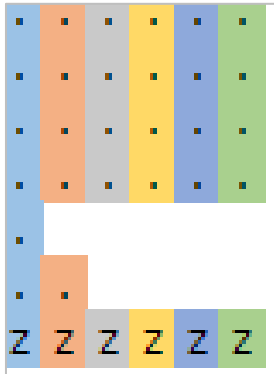
●詳細
・氏名は重複を許可する。
・メールアドレスは重複を許可しない。
・タイムゾーンの一覧は moment-timezone ライブラリの moment.tz.names() を用いて取得する。
・タイムゾーンはアルファベット昇順でソートして表示する。

★[田中:2025/11/19] moment-timezone は Moment.js と同様非推奨になったのでは？
[山田:11/20] 調べます。Luxon や Day.js の代用も検討します。

A～F列で各行ごとに
確認・実装・評価状況を管理

★の個数が残課題数

設計仕様書は評価仕様書だ



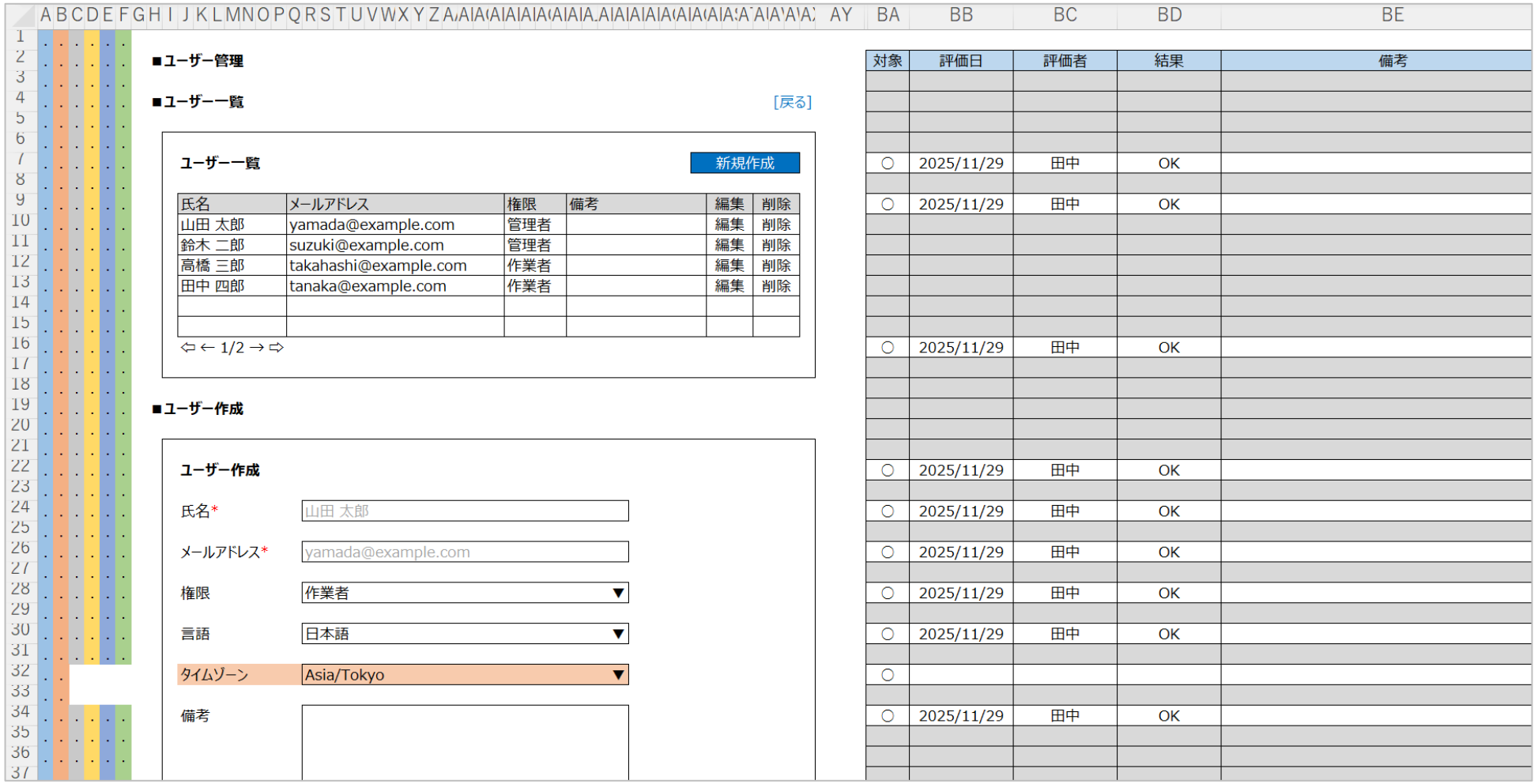
●詳細
・氏名は重複を許可する。
・メールアドレスは重複を許可しない。
・タイムゾーンの一覧は moment-timezone ライブラリの moment.tz.names() を用いて取得する。
・タイムゾーンはアルファベット昇順でソートして表示する。

仕様書の末尾に「～か？」をつけると評価仕様書となるように記載していく。

～を用いて取得するか？
～昇順でソートして表示するか？

設計仕様書をそのまま評価仕様書に

- シートの左側を 設計仕様書、右側を 評価仕様書 として使用



■ユーザー管理

■ユーザー一覧 [戻る]

ユーザー一覧 新規作成

氏名	メールアドレス	権限	備考	編集	削除
山田 太郎	yamada@example.com	管理者		編集	削除
鈴木 二郎	suzuki@example.com	管理者		編集	削除
高橋 三郎	takahashi@example.com	作業者		編集	削除
田中 四郎	tanaka@example.com	作業者		編集	削除

⇦ ← 1/2 → ⇨

■ユーザー作成

ユーザー作成

氏名* 山田 太郎

メールアドレス* yamada@example.com

権限 作業者 ▼

言語 日本語 ▼

タイムゾーン Asia/Tokyo ▼

備考

対象	評価日	評価者	結果	備考
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○	2025/11/29	田中	OK	
○				
○	2025/11/29	田中	OK	

課題管理表もまだExcelを使うことがあります…

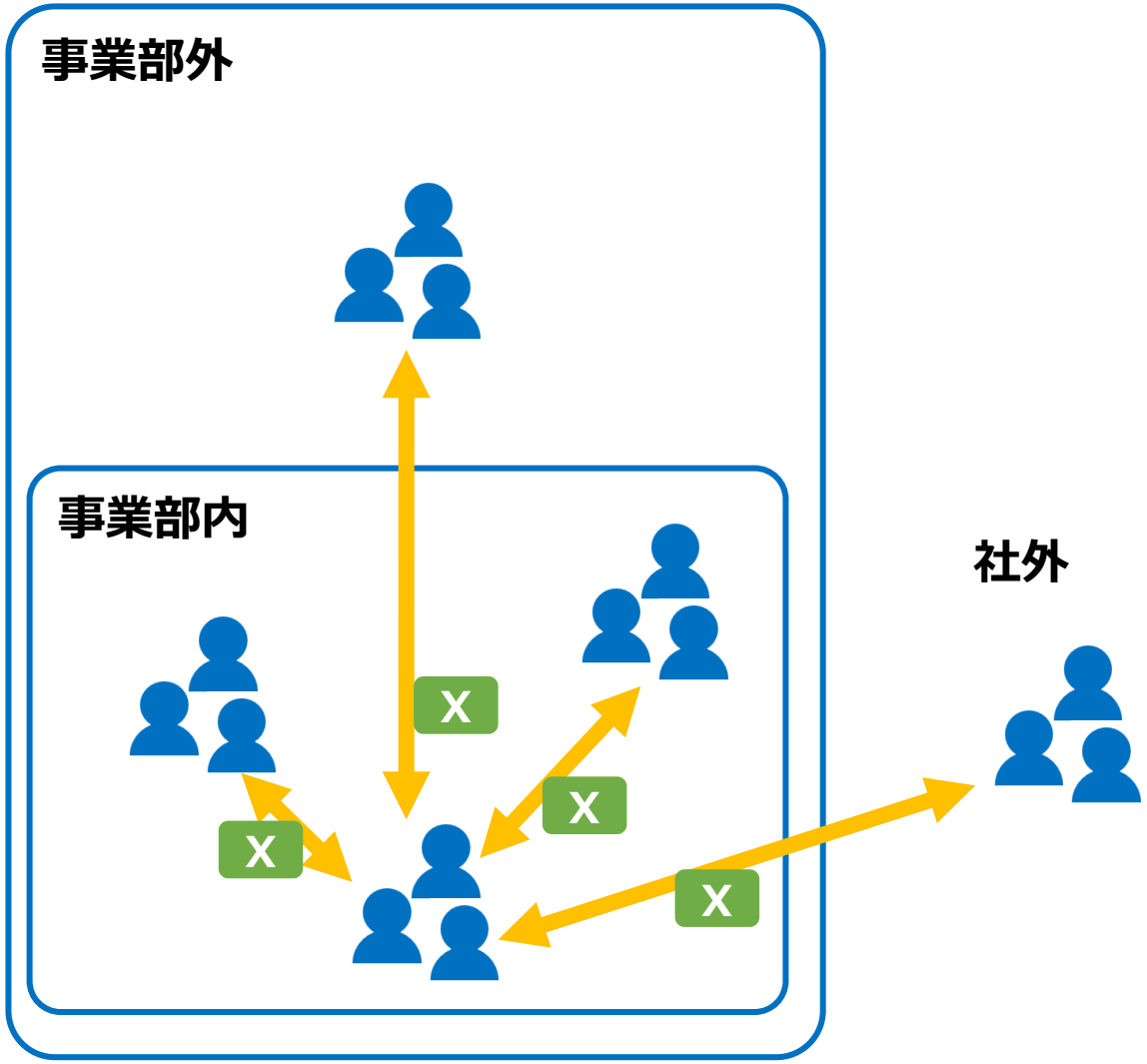
■課題管理表

記入日		記入者	内容	バトン		回答	備考
2025/11/01	●	鈴木	第2評価環境で upload コマンドを実行すると No such file. エラーが出ました。	鈴木	[山田:11/2] [鈴木:11/2] [山田:11/2] [山田:11/3] [鈴木:11/3]	.env ファイルが作成されていないのではないでしょうか。 .env ファイルはあります。 ログファイルがまだ存在しない場合にエラーを出していました。修正します。 修正しました。エラーメッセージにはファイル名も表示するようにしました。 動作を確認しました。	
2025/11/02		鈴木	upload コマンドで転送先のフォルダがあふれていた場合 System error になります。適切なエラーメッセージに修正できますか。	山田	[山田:11/3]	修正します。少々お待ちを…。	

回答者欄・回答日欄は廃止しました
代わりに バトン(担当者)欄 を追加

Box 上の Excel だと
他の人の編集もリアルタイムで参照できるので、
Excel上でチャットが始まることも…

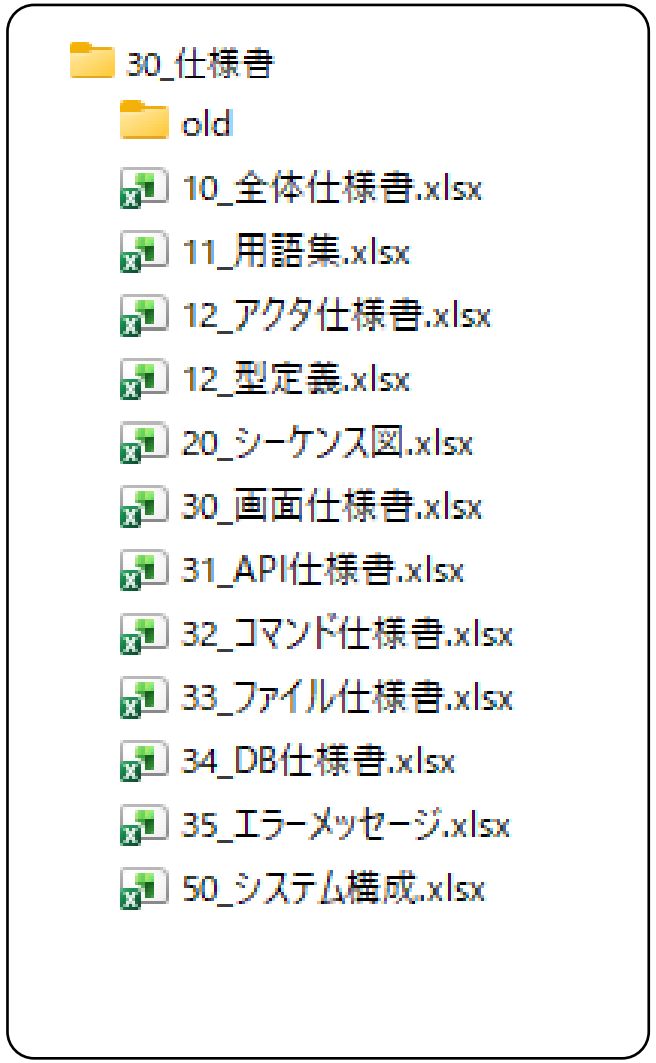
今時Excel…？



事業部内外・社外など様々なチームと
連携・調整が必要で、アクセス権などの
問題もあり、なかなかプロジェクト
管理をひとつのシステムに集約できない(泣)

Backlog などにシステム統一
できればいいのに…

仕様書フォルダ



仕様書フォルダの中身はだいたいこんな感じ。

概要設計、基本設計などには分かれていません。

ファイル名の日付(YYYYMMDD)はやめました。

（HTML仕様書などもバージョンレスの時代）

型定義は大事

■型定義

カテゴリ	型名	説明	JSON	MariaDB	備考
真偽値	Bool	・Boolean。 ・0～2^32 までの整数。 ・MariaDB の SIGNED INT に格納できる正の整数。	true / false	BOOLEAN NOT NULL	
数値	UInt4	・Unsigned Integer(4 bytes)。 ・0～2^32(約42億)までの整数。	number	INT SIGNED NOT NULL	・常に Sined か Unsigned か、4バイトで足りるか否かを意識する。何も考えず使ってしまう Int の様な型名は定義しない。
	UInt8	・Unsigned Integer(8 bytes)。 ・0～2^64(約1,844京) までの整数。	number	BIGINT SIGNED NOT NULL	
	SInt4	・Signed Integer(4 bytes)。 ・-2^31～2^31-1(約21億) の整数。	number	INT SIGNED NOT NULL	
	SInt8	・Signed Integer(8 bytes)。 ・-2^63～2^63-1(約922京) までの整数。	number	BIGINT SIGNED NOT NULL	
	Price	・価格(円)を示す数値。 ・整数部：1～10桁、・小数部：0～4桁。 ・例：1234567890.1234	string	DECIMAL(14,4) NOT NULL	・価格は JSON では文字列として扱うこと。
時刻	Date	・日付。 ・YYYY-MM-DD 形式。 ・2025/3/2 のような0埋めしない形式も許可する。	string	DATE NOT NULL	
	Time	・時刻。 ・HH:MM[:SS] 形式。 ・3:00:00 のような0埋めしない形式も許可する。 ・3:00 のような時分のみの形式も許可する。	string	TIME NOT NULL	
	DateTime	・日時	string	DATETIME NOT NULL	
文字列	AString SString UString	・Unicode String。 ・改行禁止文字列。 ・Unicode で定義される文字の内、C(その他制御文字)、Z(区切り文字)を除く、L(文字)、M(記号)、N(数字)、P(句読記号)、S(記号) に属する文字。 ・サロゲートペア領域の文字を含む。	string	VARCHAR(n) NOT NULL	

型名をしっかり定義しておくことをお勧めします。

- ・改行を許すか
- ・制御文字を許すか
- ・負数を許すか
- ・サロゲートペア領域を許すか
- ・日付の0埋め無しを許すか
- ：

余談：キーボードから「3」、「１」と入力すると…

エクスプローラーで左記が
表示されている場合、
キーボードから
「3」、「1」と入力すると
どのフォルダが選択される？

・3　1 だと 11_要件定義
・3 1　だと 31_内部結合テスト

開発ガイドライン

■設計

大分類	中分類	小項目	説明	備考	記入日	記入者	鈴木	田中	山田	佐藤	太田	高橋	渡辺	吉田
仕様書	仕様書	改造仕様書	バージョンに依存する仕様書は「改造仕様書」として扱い、このバージョンでどのような改造を行うのかを記載すること。また、出荷後は「改造仕様書」はゴミとして扱い、仕様書として残すべき仕様はすべて「最新仕様書」に記載すること。改造仕様書は [10_開発]-(作成年)-(バージョン) フォルダ配下に格納すること。		2025/11/29	山田	○	○	○	○	○	○	○	○
		最新仕様書	バージョンに依存しない最新版の仕様書として「最新仕様書」を作成・維持し、常に最新の仕様書を漏れなく・重複なく記載すること。最新仕様書は [30_仕様書] フォルダ配下に格納すること。		2025/11/29	山田	○	○	○	○	○	○	○	○
		用語集	プロジェクト毎に「用語集」を作成・維持すること。ひとつの対象に対してプロジェクト内で複数の表現が出現した場合、用語集に登録し、正式名称と許される別称を明確に定義すること。		2025/11/29	山田	○	○	○	○	○	○	○	○

- 開発における注意事項を [管理] [設計] [開発] [評価] [出荷] などのフェーズ毎に約500項目のガイドラインとして整備。
- 新規に追加した項目は、メンバ全員が確認し、理解したマーク(○)をつけていく。
- ノウハウを**集約**し、**伝播**させ、**継承**していく。

開発ガイドラインの例

- 管理
  - パスワード等の秘匿情報は ○○○.xlsx のみに記載し、他のファイルには記載しないこと。
- 設計
  - K や M などは 1,000 か 1,024 かを明確にするためのルールを決めること。
  - 非機能要件設計はデジタル庁の「地方公共団体情報システム非機能要件の標準」を参照すること。
- 開発
  - Python の四捨五入関数 round(2.5) は 3 ではなく 2 を返すことに注意。（銀行丸め）
  - Python の正規表現 ^[a-z]+$ は改行コードにもマッチしてしまうことに注意。
- 評価
  - 静的脆弱性検証（Bandit, SonarQube等）を実施すること。
  - 動的脆弱性検証（Nessus, OWASP ZAP等）を実施すること。
- 出荷
  - 出荷判定では誰がどういう基準で判定したかの記録を残すこと。
- 過去障害
  - 三値問題：DBカラムが文字列、空文字、NULL の三値を取りえたため…

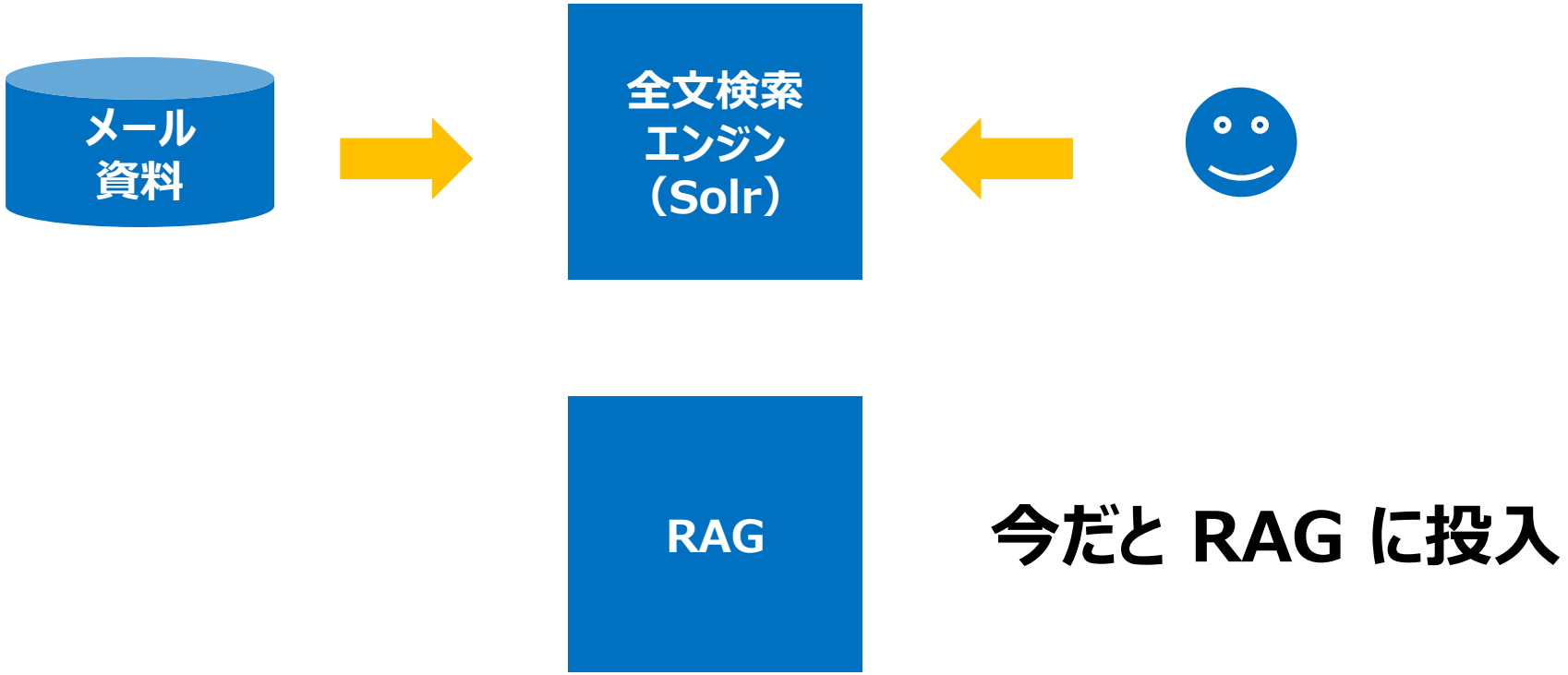
障害
↓
解決
↓
なぜ3
↓
開発ガイドライン
↓
教育
↓
継承

その他昔やっていた施策・・・

- 全文検索
- プログラミングスキルの見える化
- オフショア開発の日本語スキル向上

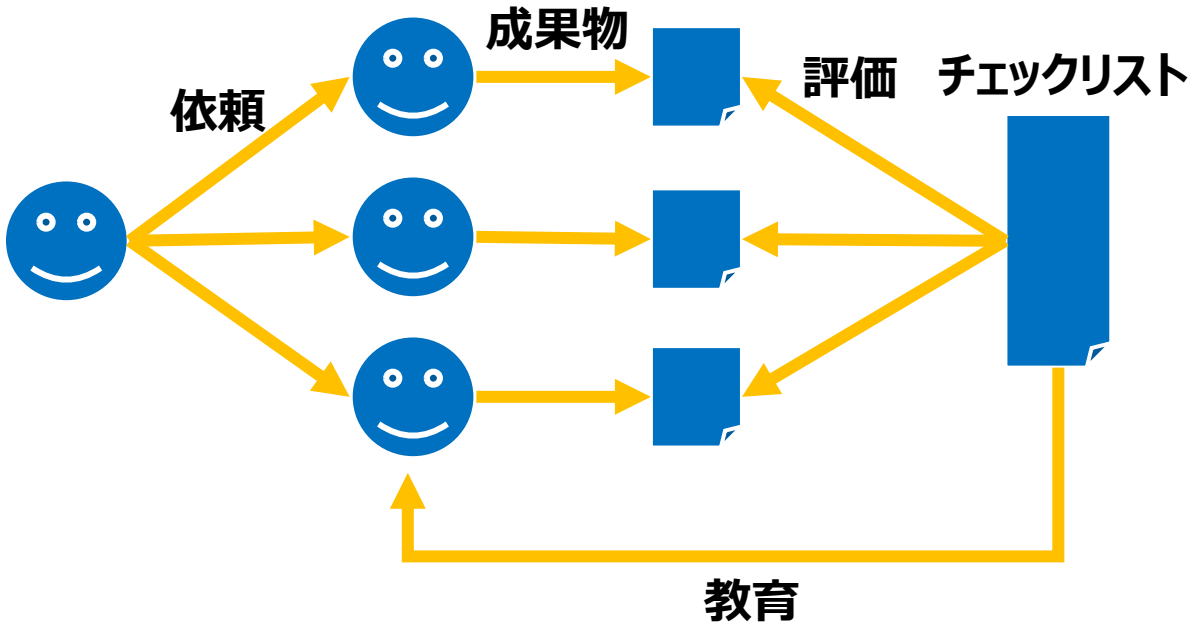
全文検索

- 過去10～20年分のメールや資料をすべて全文検索エンジンに投入
- お客様からの問い合わせ対応に活用



プログラミングスキルの見える化

- Linux の cat コマンドの互換コマンド mycat をC言語で作成してください
- オプションは不要です

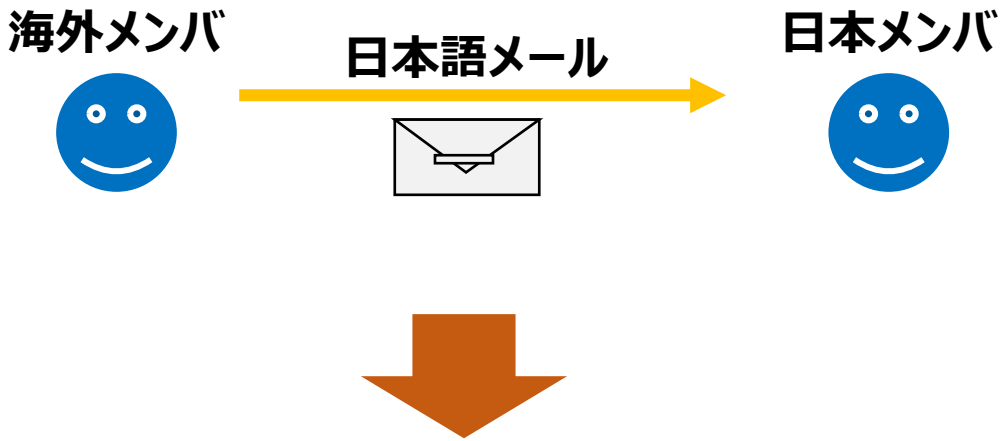


- 正常に動作するか
- エラーメッセージは適切か
- 複数ファイルを処理できるか
- 引数が無い場合は標準入力から読み込むか
- 正常時の終了ステータスは 0 か
- 異常時の終了ステータスは 0 以外か
- 読込もうとしたバイト数より読込めたバイト数が少ない時のことを考慮しているか
  → fread()やread()の場合は正常動作
- 書込もうとしたバイト数より書込めたバイト数が少ない時のことを考慮しているか
  → fwrite()の場合はエラー
  → write()の場合は正常動作
- close()やfclose()のエラーを確認しているか
- fxxx()系エラー処理にperror()を使用していないか

プログラミングスキルを客観的に**測定**。**教育**する

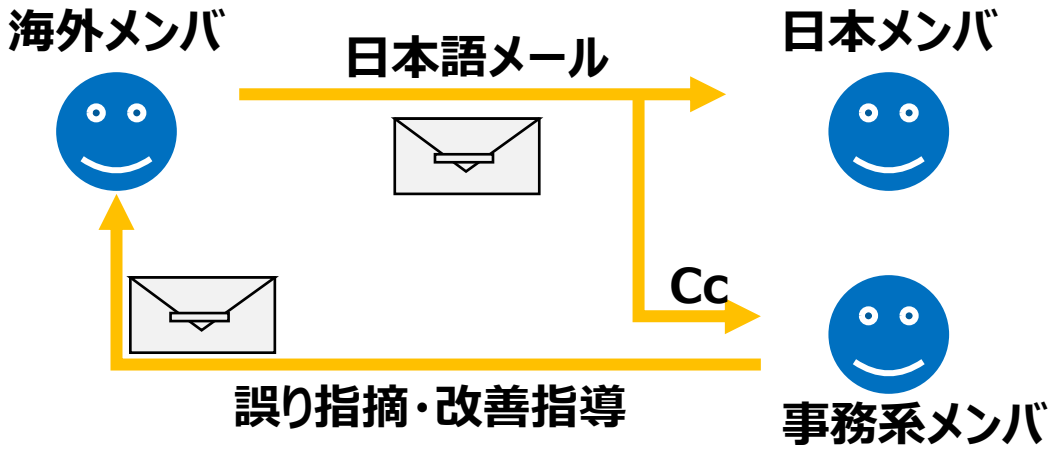
オフショアメンバの日本語スキル向上施策

改善前



- 多少日本語が変でも指摘するのが面倒
- 指摘した場合、お礼メールの返信も必要

改善後



- 事務系メンバにもメールを Cc: 送信
- 事務系メンバは日本語の誤りや改善を指導
- 指摘・指導メールへのお礼は不要とする
- 今なら AI にやってもらってもよし

改善活動の継続

- こうした改善活動を整理して他のプロジェクトと**共有**
- 年間約60個の改善施策を実施
- 改善施策の内容自体を次の時代のメンバに**継承**

10年後のメンバにも伝承する

知識やノウハウを「**増幅 (AMP)**」させ、
日々の業務やプロジェクトを大きく「**飛躍 (LEAP)**」させていく

ご清聴ありがとうございました