

テスト設計コンテスト'23 決勝プレゼンテーション資料

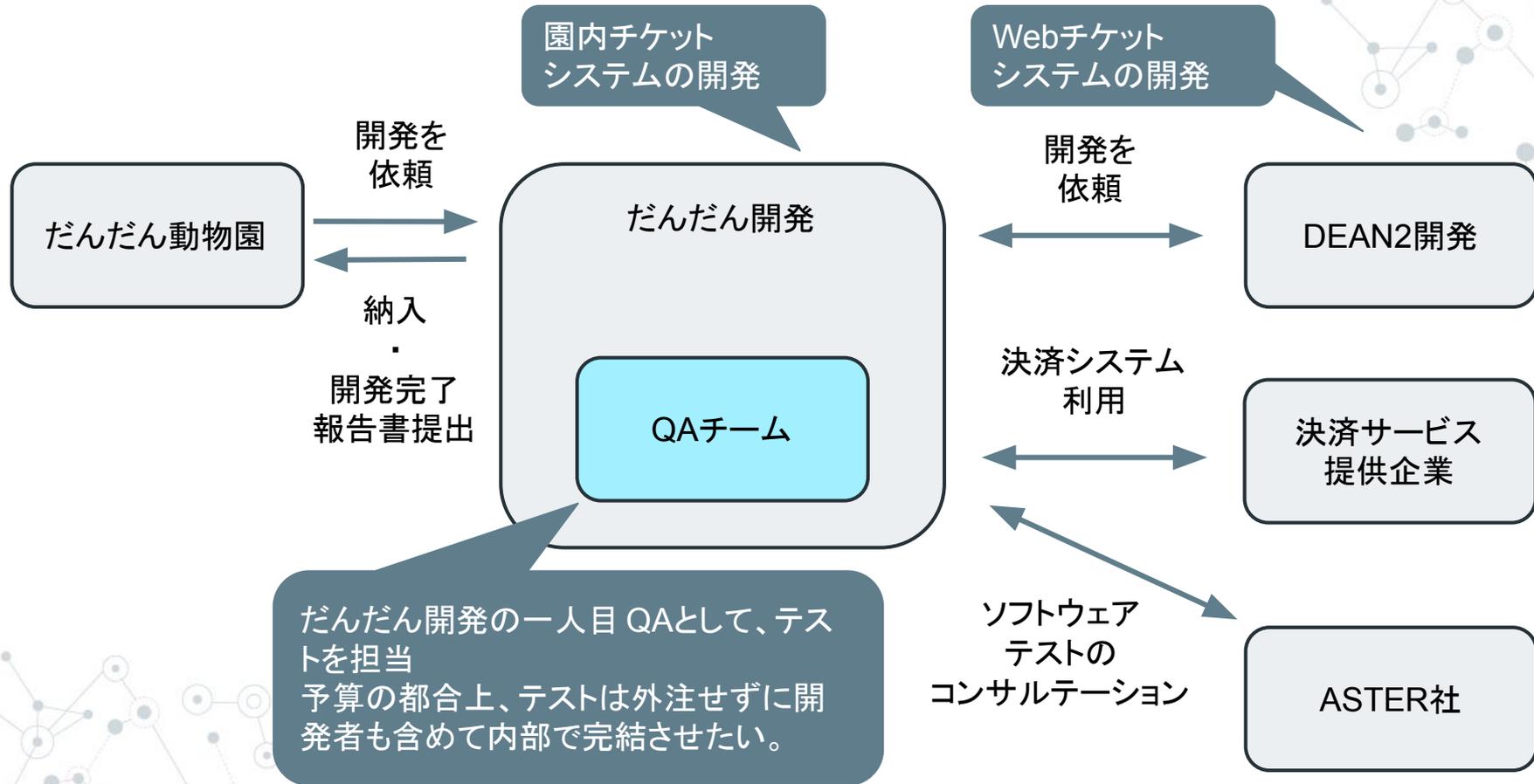
考えるアシカ



予選(第1回レビュー)からの主な変更点

- テストアーキテクチャの見直し
- リスク分析の方法を見直し
 - 『起こってほしくないこと』と『起こさないためのアプローチ』を具体化
 - ChatGPTを用いたリスク抽出
- ChatGPT × 意地悪漢字を用いたテスト要求抽出
- テスト詳細設計以降の検討

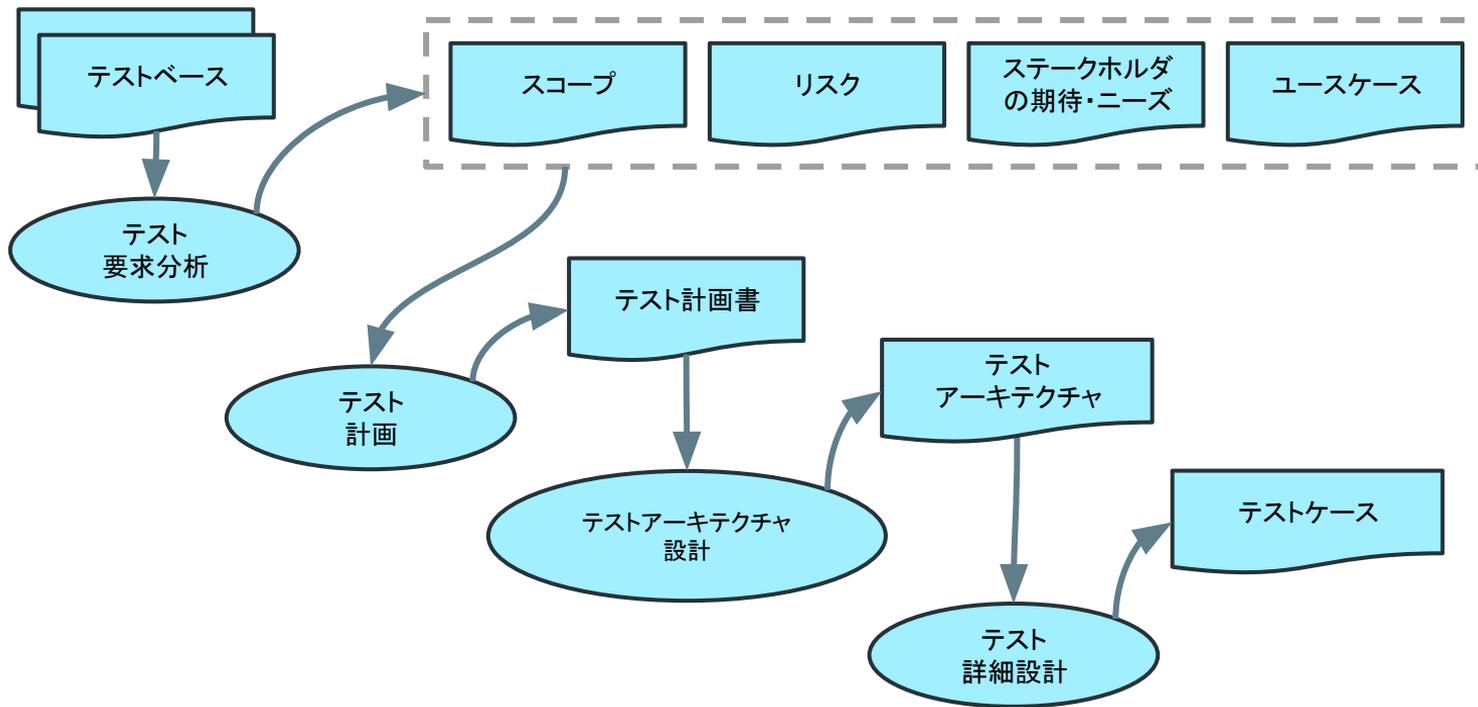
チームの位置付け



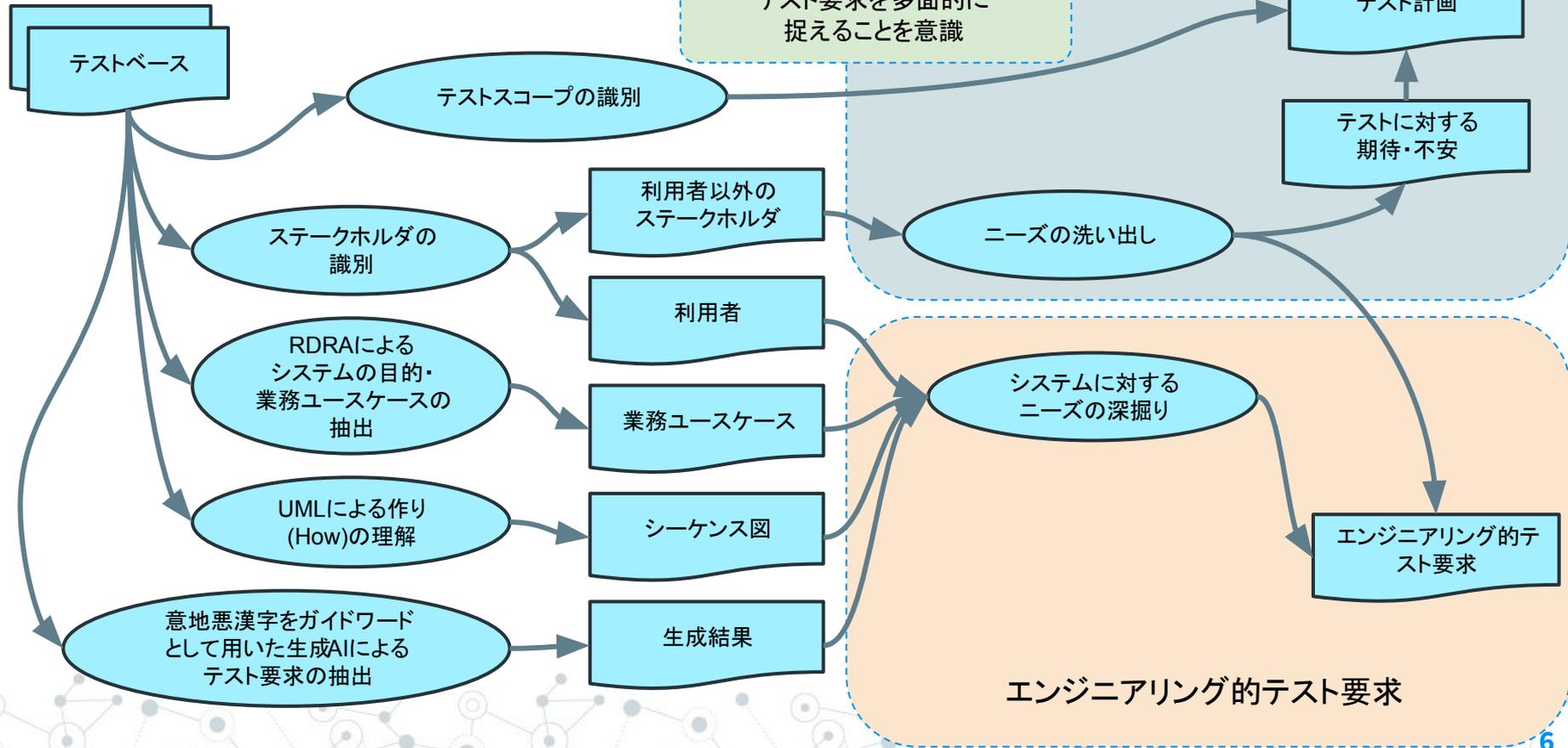
背景(想定)

- 本プロジェクトは予算の都合上、だんだん開発のQAと開発者でテストを行う
- 各サブシステムを結合した状態でのテストは、限られた期間でしか行うことができない
 - 園内チケットシステムとWebチケットシステムはそれぞれ別のベンダーで開発されており、同時に開発が進められている
- だんだん市の補助金申請期限が予め決まっている
 - 入場者が「密」にならないための入場規制が行えていることを示す開発完了報告は申請期限に間に合うよう提出する必要がある
- なにより、入園者に1日も早く「安心安全」に動物園を楽しんでもらいたい

テストプロセスの全体像



テスト要求分析 - プロセス

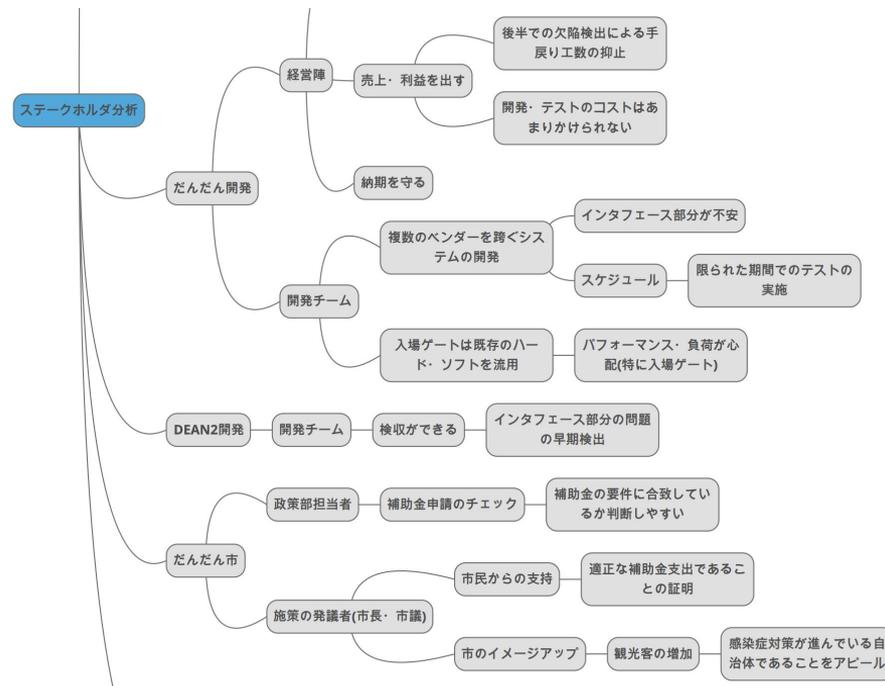


テスト要求分析 - ステークホルダの洗い出し

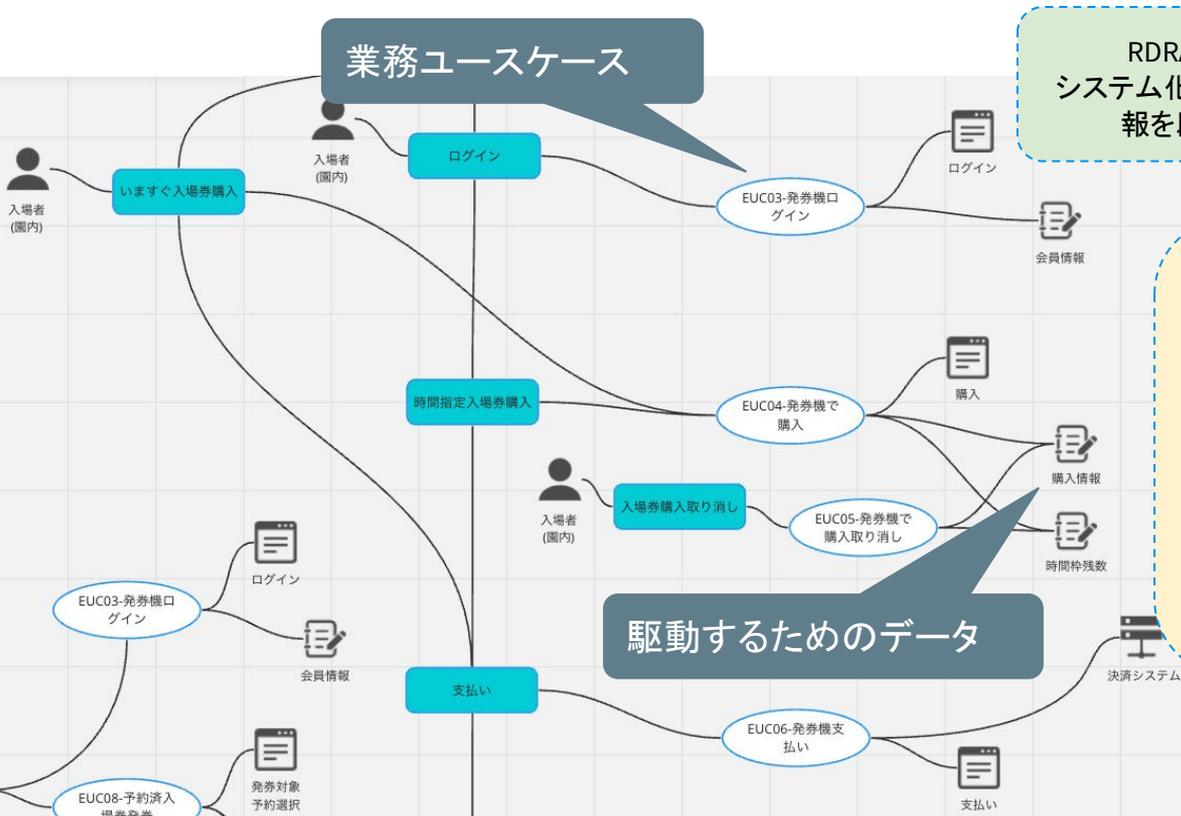
以下のような流れでステークホルダとその要求を洗い出していた

- どんなステークホルダがいるか？
- そのステークホルダの立場や役割は？
- その役割からみたシステムやテストの期待や不安は？

得られたテスト要求は、「マネジメント的テスト要求」として、テストアーキテクチャへの反映を行なった



テスト要求分析 - RDRAによるモデリング



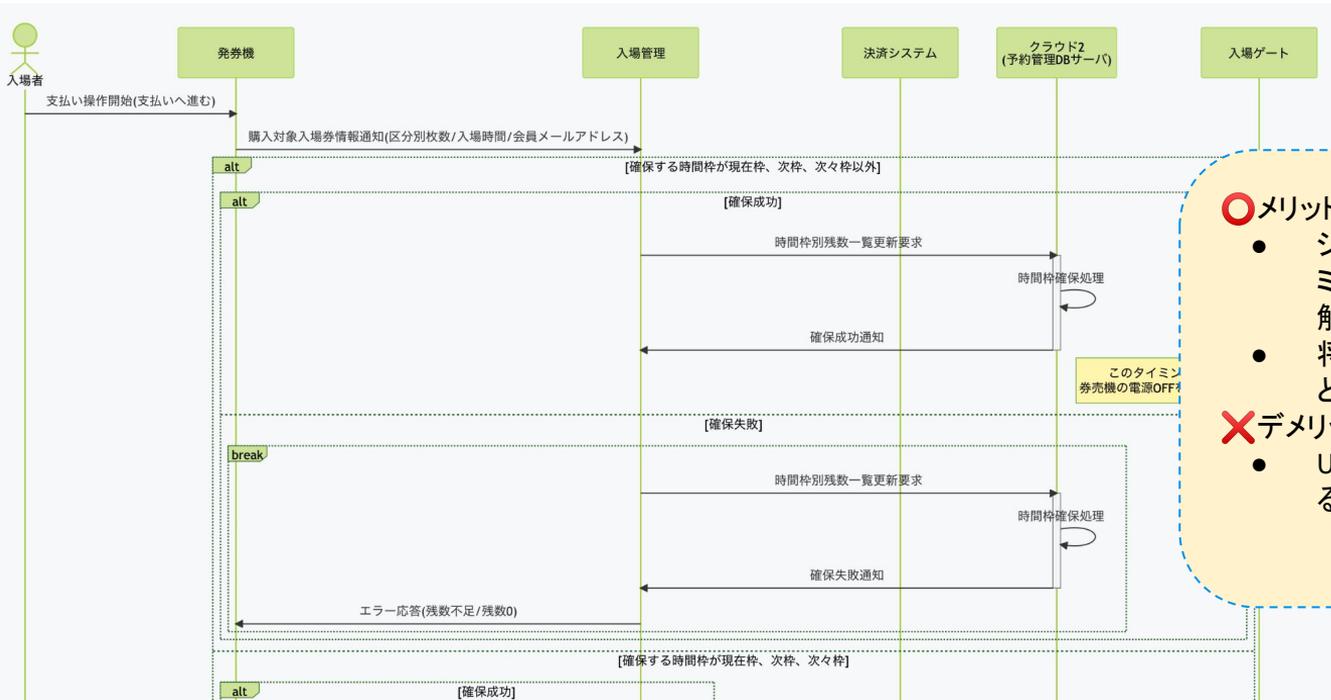
RDRA2.0は要件定義を行うためのフレームワーク。
システム化の目的から業務の内容、それを駆動するための情報を段階的に詳細化しながらモデルで表現できる。

- **メリット・効果**
 - 仕様書の書き方によらない理解
 - 機能間の関連を意識したテスト要求の抽出
 - 後のテストアーキテクチャでの活用によるテスト設計成果物の一貫性への寄与
- ✕ **デメリット**
 - システムの設計方法など Howは関心事として扱わない

ユースケース複合図(業務フローつき)の一部

テスト要求分析 - システムの作りを理解

RDRAで表現されないシステムの作り (How)を理解するため、UML(シーケンス図)を作成！



○メリット・効果

- システム構成要素間のデータの流れ・タイミングを図示したことで、テスト対象への理解が深まった
- 将来的な拡張を見据えた時のボトルネックとなりうる箇所を推測するのに役立った

✗デメリット

- UMLにより直接的にテスト要求が抽出できるわけではない

作成したシーケンス図の一部

意地悪漢字をガイドワードとして用いた生成AIによるテスト要求の抽出

鈴木 三紀夫氏の『意地悪漢字』[*1]
をガイドワードとして与え、
生成AIによるテスト要求の抽出を実
施。(モデルはGPT4)

使用したプロンプト

意地悪漢字という考え方があります。
意地悪漢字は、テスト設計において、意地悪テストを考える時のガイドワードとして利用することができます。

意地悪漢字を用いて、システムの欠陥を見つけるためのテストケースを作ってください。
結果には選択した意地悪漢字、テストケースの概要、根拠を箇条書きで10個あげてください。
根拠には必ず意地悪漢字の一文字が含まれるようにしてください。

意地悪漢字としては以下が挙げられます。

- 落、除、捨、解、略、欠
- 止、失、無、減、消、廃
- 稀、異、違、不、未、誤
- 降、急、延、退、遅、縮
- 壊、切、減、衰、断、禁
- 飛、移、隠、乱、散、迷
- 間、中、外、周、端、限
- 戻、代、替、変、再、返
- 共、連、続、合、同、交
- 別、分、割、離、脇、入
- 優、特、過、極、越、超
- 全、片、独、差、微、満

対象となるシステム・機能は次のとおりです。
{テストベースの内容を記述する}

*1 鈴木 三紀夫: 意地悪漢字, JaSST2010 Hokkaido

意地悪漢字をガイドワードとして用いた生成AIによるテスト要求の抽出

生成結果例(一部)

○メリット・効果

- テスト設計初学者が、テスト条件を抽出するための参考にできる
- 経験者でも漏れていたテスト条件に気づくことがある

✕デメリット

- 出力結果をそのまま活用することはできない
- 数撃って当てる必要がある

具体例を示してもらえることで、漢字1文字よりは発想を広げやすい

1. 漢字: 欠

- テストケース: メールアドレスやパスワードに必要な文字が欠けている場合のログイン試行
- 根拠: 欠けた情報の扱いを確認する

2. 漢字: 無

- テストケース: ログイン時にメールアドレスやパスワードを無入力にする
- 根拠: 無入力状態の扱いをテストする

3. 漢字: 誤

- テストケース: 会員情報と異なる誤ったメールアドレスやパスワードでのログイン試行
- 根拠: 誤入力時のシステムの反応を確認する

(～中略～)

6. 漢字: 限

- テストケース: 同時ログイン試行者数がシステム限界を超えた場合
- 根拠: システムの負荷耐性をテストする

7. 漢字: 消

- テストケース: システムから会員情報が突然消えた場合の対応
- 根拠: データ損失時のエラー処理を検証する

8. 漢字: 間

- テストケース: 時間枠の間に変更が生じた場合のシステム対応
- 根拠: 時間枠更新時のデータ処理を確認する

9. 漢字: 超

- テストケース: 入場券の購入枚数が許容枚数を超えた場合
- 根拠: 超過状況でのシステムの制限を検証する

10. 漢字: 変

- テストケース: 会員情報(メールアドレスやパスワード)が頻繁に変更される場合
- 根拠: 情報変更頻度が高い場合のシステムの安定性を検証する

リスク分析

リスクはテストベースからだけでなく、UMLや生成AIからも多面的に抽出

想定されるリスクを次の4つに分類。
リスクがどのような影響を及ぼすのかを一目でわかるように！

- **事業におけるリスク**
→ 売上や利益など
- **運用におけるリスク**
→ 動物園スタッフなど
- **利用におけるリスク**
→ 動物園の入園者など
- **セキュリティにおけるリスク**

リスクの軽減策は『テストに盛り込んだり、強化する』だけでなく、**設計へのフィードバック**を行うことで、より早いタイミングでのフィードバックを実現！

想定されるリスク	リスクの種類	発生確率	インパクト	算出したリスクレベル	リスク対策種別	リスク回避・軽減の方法	リスク抽出元
QRコードに埋め込まれた不正な構文(SQLインジェクション)により、秘匿化されるべきデータを表示できる懸念はないか。	セキュリティにおけるリスク	2	3	6	設計フィードバック	入力された文字列をサニタizingする処理が入っていない場合は追加いただく。	テストベース
悪意を持ったユーザーが何回もログインを試行してアカウント情報を窃取する操作を防止できているか。	セキュリティにおけるリスク	2	2	4	設計フィードバック	制限なくログインが試行できる仕組みになっている場合は、今後のエンジニア対応として、ログイン回数制限を設けることを検討いただく。	テストベース
「11_園チケットシステム要求仕様書.pdf」「P21」「EK-210 時間枠別残数情報配信」より、発券機からの配信要求タイミングとして、以下の記述がある。 >開始画面(S-001-01)表示中に入場者が任意の操作実施時 もし、定期的な時間枠別残数情報を行っていない場合、入場者が操作を行わない限り残数情報が更新されないため、前の時間枠で時間枠がなくなった場合、次の時間枠に変わっても残数なしと表示され、本来購入できる入場券が購入できないと勘違いされる懸念がある。	事業におけるリスク	2	1	2	設計フィードバック テスト強化	開始画面の時間枠別残数情報の取得タイミングについて確認いただく。 もし、時間枠別残数情報を入場者の操作実施時しか行っていない場合は、定期的な情報取得を検討いただく。 また、時間枠別残数情報の取得・残数による遷移可否の処理については、テストのカバレッジを強化する。	テストベース
新しい時間枠の開始時に行われる処理が複数存在し、ハードウェアの処理速度やネットワークの帯域の不足などにより、処理が遅延する可能性がある。 また、将来的に時間枠残数を増やし、購入可能な枚数が増えたとときに、システムの負荷となる可能性がある。 - 入場管理の時間枠残数取得処理 - 入場管理で管理を引き継ぐ時間枠(次枠、次々枠、さらにその次の枠)の時間枠別残数情報の取得 - 入場ゲートの入場券情報一覧更新 - 状態が未使用のままだった入場券情報を無効にし、Webチケットシステムの予約管理DBサーバに送信する処理 - 新しい時間枠分の入場券情報一覧を同期する処理	システムにおけるリスク	2	3	6	設計フィードバック	時間枠残数の上限まで購入した状態でのテストを実施し、処理に遅延が発生しないかを確認するボリュームテストを実施する。 また、将来的なスケール(時間枠残数の増加)に耐えられるようシステムアーキテクチャの見直しを検討いただく。 (例えば、時間枠別残数情報を新しい時間枠の開始前に廃棄し完全に書き換え、新しい時間枠の開始時にはその差分のみを取得することで負荷を低減させることを検討いただく)	テストベース
上記と同様に、いままぐ入場券の購入が同時に行われた場合に、ハードウェアの処理速度やネットワークの帯域の問題で、入場券情報の同期処理が遅延する可能性がある。	システムにおけるリスク	1	2	2	テスト強化	発券機から同時にいままぐ入場券を購入した際、入場ゲートへの入場券情報同期処理が遅延しないことを確認するテストを追加する。	UML
途中まで金額を投入後、購入を取り消した場合、投入した金額が返却されたことに気づかず立ち去ってしまう可能性がある。	利用におけるリスク	2	2	4	設計フィードバック	取消ボタンを押下後、すぐに画面遷移させず、先に現金を払い出してから、数経過後に初期画面に遷移することを検討いただく。	テストベース
システム構成要素の1つで障害が発生すると全機器が使用できなくなる仕様となっており、園内が混雑する可能性がある。	システムにおけるリスク	1	2	2	設計フィードバック	予め購入しておいた時間指定入場券については、新しい時間枠の開始時、入場ゲートに入場券情報が同期されているため、入場させることができる可能性がある。 障害の発生箇所に応じて、稼働を継続するシステム構成要素を判断するアーキテクチャを考慮いただく。	テストベース
30分以上障害が継続した場合、購入した入場枠で入場できない可能性がある。	運用におけるリスク	1	2	2	設計フィードバック	障害発生時、動物園の係員が手動で入場するオペレーションを考慮した作りになっているか確認いただく。	テストベース
ボタン制御が不十分で、過去の時間枠が選択できてしまう、画面操作中に時間枠が変化したのが、画面上では開始後の時間枠が選択できてしまう可能性がある	システムにおけるリスク	2	2	4	テスト強化	販売終了した時間枠が選択できないことを確認するテストを実施する。	生成AI

応用HAZOPのガイドワードを用いたリスク項目の抽出

多面的な視点でリスク分析を進めるため、生成AIに適用HAZOPのガイドワード[*2]をプロンプトとして与え、リスク項目の抽出を試行。(モデルはGPT4)

使用したプロンプト

まずは、次のテーブルを読み込んでください。

着目点1	着目点2	ガイドワード
---	---	---
振る舞いそのもの有無	全く～しない	
振る舞いそのもの程度	強く、弱く	
振る舞いそのもの速度	急いで、ゆっくり	
振る舞いそのもの持続時間	ずっと、短く	
振る舞いそのもの範囲	余分に、不十分に	
振る舞いそのもの向き	反対に、他に	
振る舞いそのもの種類	違う	
振る舞いそのものタイミング	遅く、早く、同時に	
振る舞いそのものタイミング	別々に	
振る舞いそのもの順序	前に、抜かして、後	
振る舞いそのもの順序	余分に、繰り返し、反対に	
振る舞いそのもの回数	多く、少なく	
振る舞いの対象対象物	違うもの	
振る舞いの対象対象物の向き	反対に	
振る舞いの対象対象物の量	多く、少なく、なし	

このテーブルはソフトウェアのリスク項目を洗い出すためのガイドワードです。

このガイドワードを用いて、与えられた使用から推測されるリスク項目を洗い出してください。

定義されたガイドワードは許可なくあなたが変更することはできません。
仕様から読み取ることで想定できるリスクについては列挙する必要はありません。
仕様に明示されていないリスクを洗い出すことが今回の目的です。

抽出したリスク項目を簡条書きにして出力してください。

より多くのリスクを抽出できるよう努力してください。

各リスク項目の出力結果には、次の点を記述するものとして、指定された出力は行わないようにご注意ください。

- ガイドワードから導出したリスク項目
- 選択した着目点1, 着目点2, ガイドワード(必ずガイドワードを記載すること)
- リスクを抽出した理由

仕様は後ほど与えられますので、まずは処理内容を理解してください。

この内容に対して意見はいりません。

*2: 河野 哲也: ソフトウェア要求仕様における
HAZOPを応用したリスク項目設計法 JaSST2012

応用HAZOPのガイドワードを用いたリスク項目の抽出

使用したプロンプト

既に人手によりリスク項目として抽出済みであったが、有効な視点

生成AIのみで完結することは難しいものの、発想を広げる目的としては有効活用できる！

EH-220 入場時間枠選択の仕様を基にリスク項目を洗い出します。以下は、リスク項目とそれに関連する着目点、ガイドワード、およびリスクの抽出理由です。

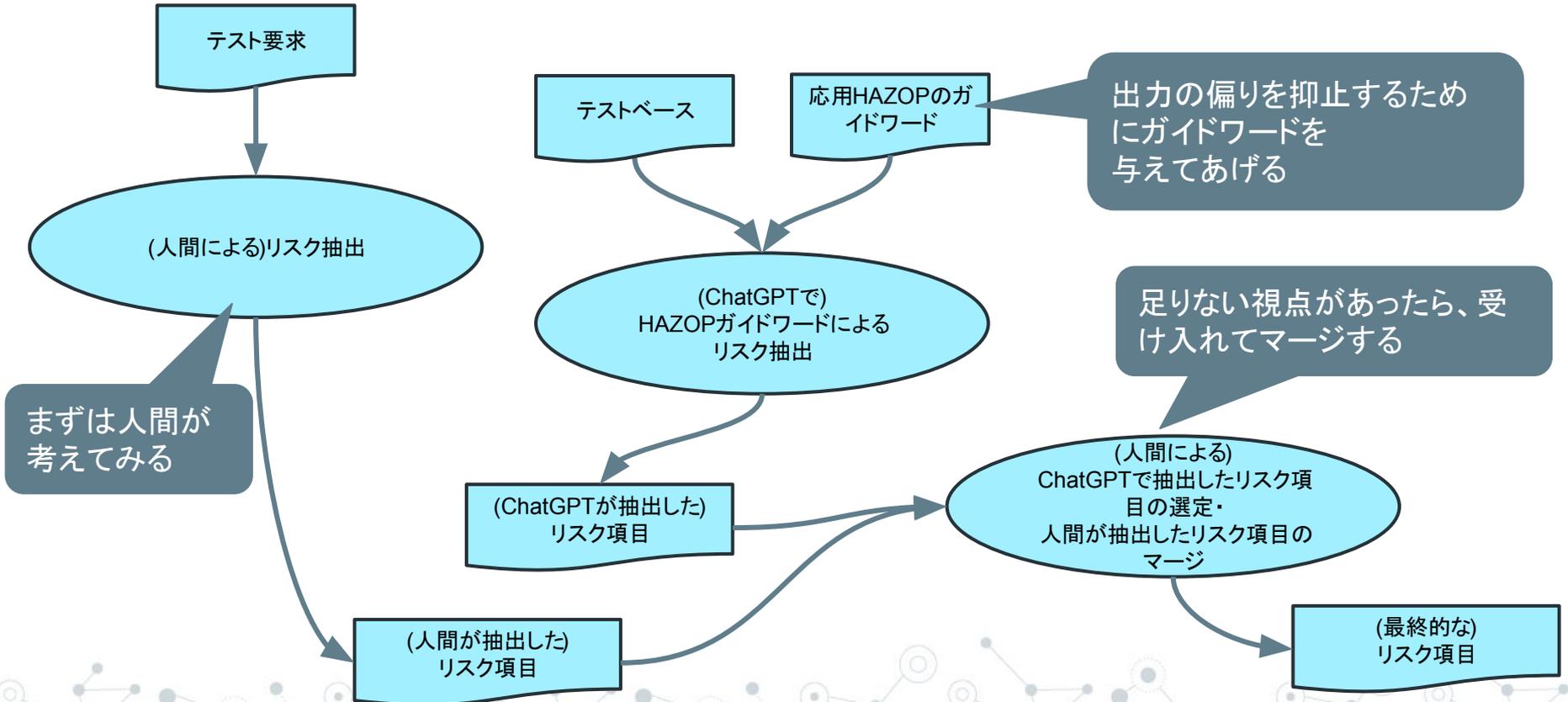
1. リスク項目: 残数情報の表示誤り
 - 着目点: 振る舞いそのもの、程度
 - ガイドワード: 強く、弱く
 - 理由: 残数が正確に表示されない場合、入場者は誤った判断をする可能性があり、これは入場チケットの誤購入や混乱を引き起こす可能性があります。
2. リスク項目: 残数情報の更新遅延
 - 着目点: 振る舞いそのもの、タイミング
 - ガイドワード: 遅く、早く
 - 理由: 情報更新が遅れると、入場者が利用可能な時間枠を正確に把握できず、適切な時間枠の選択に影響を与える可能性があります。

(～中略～)

5. リスク項目: 残数表示「○」「数字」「x」「ー」の不適切な切替
 - 着目点: 振る舞いそのもの、タイミング
 - ガイドワード: 遅く、早く
 - 理由: 残数表示の切替が適切なタイミングで行われない場合、入場者が誤った情報に基づいて決定を下す可能性があります。

これらのリスク項目は、仕様に明示されていないが、ガイドワードを用いた分析により抽出されたものです。発券機の運用において、これらのリスクを考慮し、適切な対策を講じることが重要です。

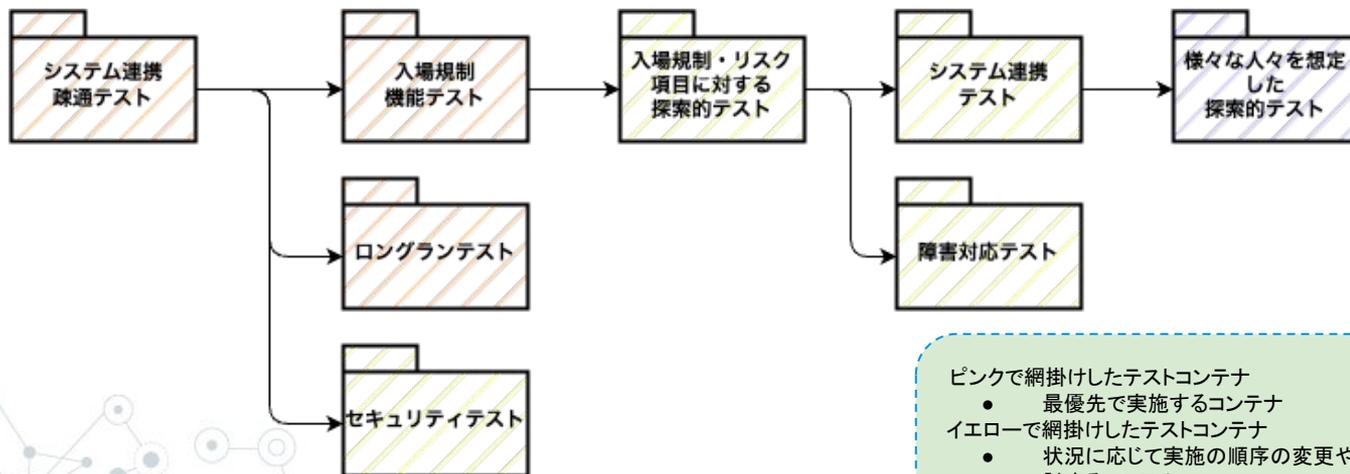
生成AIをテスト活動に用いる場合の考慮点 - リスク分析の例



テストアーキテクチャ設計

テストアーキテクチャにおいて考慮したこと

- システム構成要素間でのコミュニケーションの問題を事前に洗い出す
- 入場規制に関わる機能が実現できていることを早期に示す
- 実施に時間のかかるロングランテストを早期に実施する



ピンクで網掛けしたテストコンテナ

- 最優先で実施するコンテナ

イエローで網掛けしたテストコンテナ

- 状況に応じて実施の順序の変更や実施不要の判断に倒すことを検討するコンテナ

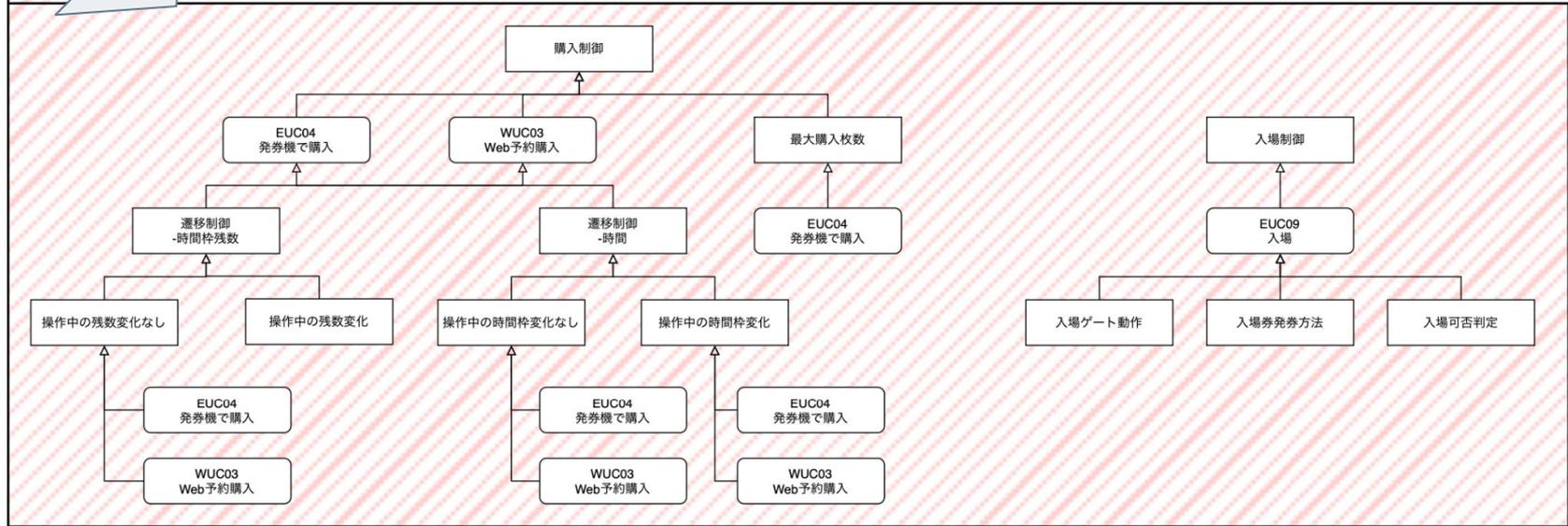
ブルーで網掛けしたテストコンテナ

- 優先度は低いが、システムとしての完成度を高めるために実施すべきテストコンテナ

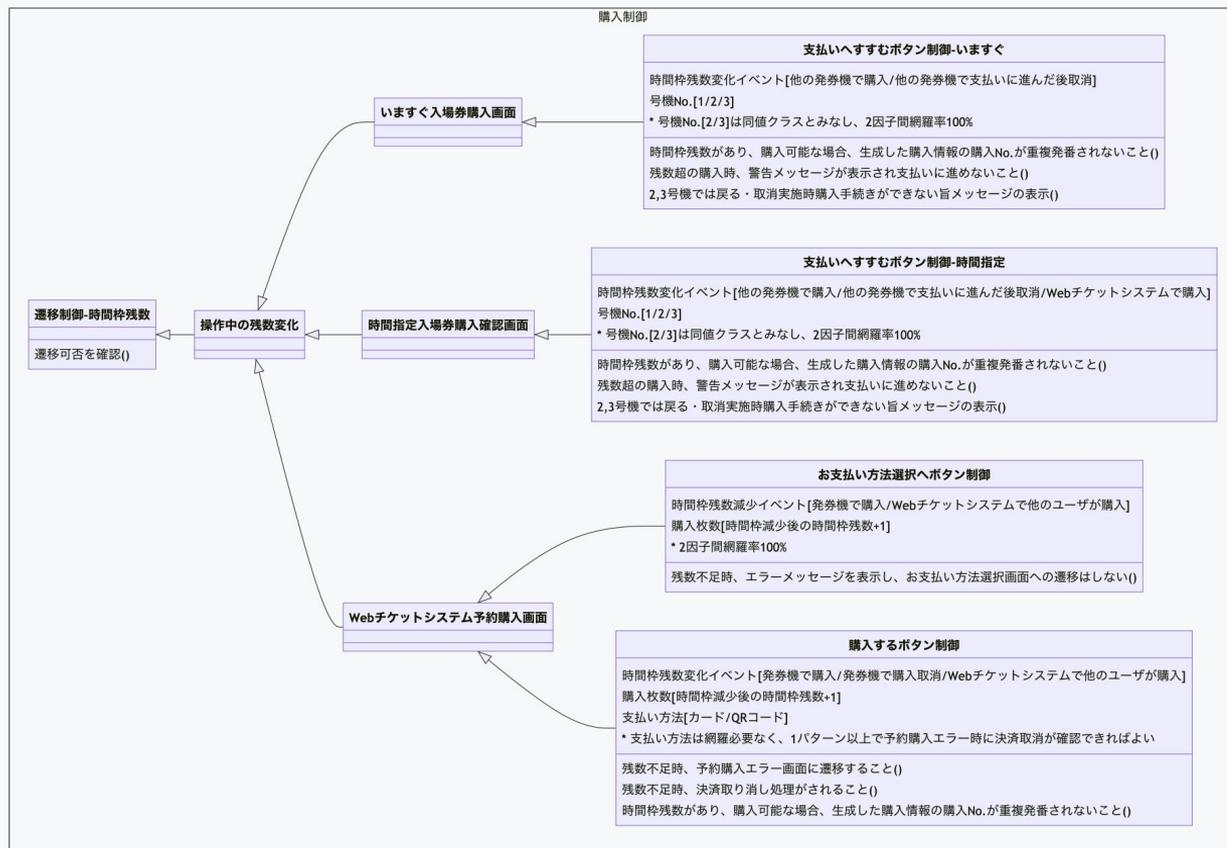
テストアーキテクチャ設計(コンテナ図)



テストコンテナの責務が
わかりやすくなるよう詳細化



テストアーキテクチャ設計(詳細図)



テストアーキテクチャと、実装結果であるテストケースとの間で、テストの意図が途切れないようにするために、テストアーキテクチャの要素毎に**テストアーキテクチャ詳細図**を作成していった

テストの責務

パラメータ1[値1/値2]

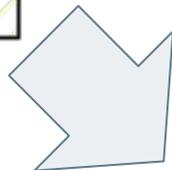
パラメータ2[値1/値2]

*網羅基準

振る舞い()

探索的テストのアプローチ(1)

入場規制・リスク
項目に対する
探索的テスト

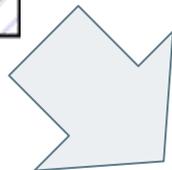


入場規制に関する機能やリスクとして抽出されたことのテストを強化するために、テストチャータを用いた探索的テストを実施

テストアイテム	どうやってテストするか	どんな不具合を発見したいか
・ WUC03-Web予約購入 ・ EUC04-発券機で購入 入～EUC07-発券	複数のシステム構成要素から同時に購入を進める。	・ 時間枠残数を越えた購入が行われてしまう ・ 購入Noが重複してしまう
・ WUC03-Web予約購入 ・ EUC04-発券機で購入 入～EUC07-発券	複数のシステム構成要素間で少しタイミングをずらして購入を進める	・ 時間枠残数を越えた購入が行われてしまう ・ 購入Noが重複してしまう
・ WUC03-Web予約購入 ・ EUC04-発券機で購入 入～EUC07-発券	発券機で支払い画面まで進んだ後、取消を行なったタイミングで、他の発券機やWebチケットシステムから購入を行う。	・ 時間枠残数を越えた購入が行われてしまう ・ 購入Noが重複してしまう
・ WUC03-Web予約購入 ・ EUC04-発券機で購入 入～EUC07-発券	発券機で支払い画面まで進んだ後、取消を行なった後、少しタイミングをずらし、他のシステム構成要素から購入を進める。	・ 時間枠残数を越えた購入が行われてしまう ・ 購入Noが重複してしまう
・ EUC04-発券機で購入	発券機で支払い画面まで進んだ後、戻る・取消を繰り返し実施する	・ 時間枠が不正に確保され続ける

探索的テストのアプローチ(2)

様々な人々を想定した探索的テスト



ユーザの特性(せっかち、のんびり、間違いやすいなど)を意識した探索的テストを実施

テストアイテム	どうやってテストするか	どんな不具合を発見したいか
全般	(せっかちな人を想定して)ボタンの二度押しや、画面遷移前に次画面のボタンを押したり、前の画面に戻るうとしたりする。特に、遷移に時間がかかりそうな箇所、具体的には通信が行われていそうなタイミング(残数取得、購入情報送信、定時処理)に着目してみる。	<ul style="list-style-type: none">・本来遷移できてはいけない画面に遷移できてしまう・残数がないのに購入に進めてしまう・状態に齟齬が生まれ、入力を受け付けなくなる
全般	(のんびりな人を想定して)ゆっくりと画面操作をしてみる。無操作タイムアウト間際の時間に操作をしてみる。	<ul style="list-style-type: none">・操作方法に慣れていない人が入力をする前に初期画面に戻ってしまう・状態に齟齬が生まれ、入力を受け付けなくなる
全般	(おっちょこちょいの人を想定して)選択内容を頻繁に変更する、前の画面に戻って選択し直す。券種を変更する、枚数を変更する、区分を変更する、時間枠を変更する、支払い方法を変更する。	<ul style="list-style-type: none">・前の選択内容が保持されている・内部的に保持している情報に齟齬が生まれ、エラーが発生する
全般	(ヘビーユーザーの人を想定して)会員情報に紐づく入場券情報が大量にある時の表示・振る舞いを確認する。	<ul style="list-style-type: none">・ページネーションが行われない・情報取得に時間がかかる

テスト詳細設計

将来的にテスト管理ツールへの移行も視野に
管理しやすいスプレッドシート形式にした

入場制限機能テスト

ID	対応するテストアーキテクチャ	テスト目的	テストアイテム	前提条件	入力	期待結果
P-001	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 10 - 号機No: 1号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入画面に遷移すること
P-002	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 10 - 号機No: 2号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入画面に遷移すること
P-003	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 10 - 号機No: 3号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入画面に遷移すること
P-004	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 9 - 号機No: 1号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入画面に遷移すること
P-005	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 9 - 号機No: 2号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-006	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 残数アイコン切替閾値: 10 - 現在時間枠の時間枠残数: 9 - 号機No: 3号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-007	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 現在時間枠の時間枠残数: 0 - 号機No: 1号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-008	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 現在時間枠の時間枠残数: 0 - 号機No: 2号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-009	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	開始画面で時間枠残数による入場券購入画面の遷移可否を確認する	開始画面	- 現在時間枠の時間枠残数: 0 - 号機No: 3号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-010	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	残数アイコン切替閾値の値により、開始画面の遷移制御が行われていることを確認する	開始画面	- 残数アイコン切替閾値: 9 - 現在時間枠の時間枠残数: 8 - 号機No: 2号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-011	操作中の残数変化なし 発券機 開始画面 > 入場券購入ボタン制御	残数アイコン切替閾値の値により、開始画面の遷移制御が行われていることを確認する	開始画面	- 残数アイコン切替閾値: 9 - 現在時間枠の時間枠残数: 8 - 号機No: 3号機	開始画面で「入場券購入」ボタンをタッチ	入場券購入ボタンに「当日入場券購入を一時停止しています」ラベルが表示され、タッチできないこと
P-012	操作中の残数変化なし 発券機 開始画面 > 時間指定入場券購入ボタン制御	当日全ての時間枠残数がなく、時間指定入場券販売画面への遷移が制御されていることを確認する	開始画面	- 検証実施時刻: 10:00-10:29の間 - 当日10:30以降の時間枠残数: あり	開始画面で「時間指定入場券購入」ボタンをタッチ	時間指定入場券販売(入場予約時間・枚数選択)(S-003-Q2)画面に遷移すること

仕様における問題点・曖昧さの検出とフィードバック

リスクに該当しない仕様書の誤記や、振る舞いとして曖昧な点についてフィードバックを行なった

質問一覧表 兼 仕様フィードバック一覧表

No	該当のドキュメント	ページ数・章節名	指摘区分	内容
7	11_園内チケットシステム要求仕様書.pdf	Z2-202	誤記	残数表示インジケータ2は、時間指定入場券の残数を画面表示する機能ですので、要求仕様書に記述の「いますぐ入場券の」という部分は不要ではないかと思われませんかでしょうか。 > 残数情報は以下の基準で切り替える。 > いますぐ入場券の残数閾値以上:○ > いますぐ入場券の残数閾値未満:△ > いますぐ入場券の残数0:×
9	①11_園内チケットシステム要求仕様書.pdf ②72_だんだん動物園入場システムデータ連携仕様書.pdf	①P21 EK-210 時間枠別残数情報配信 ②P8 6「入場管理」残数表示インジケータ2「間データ	仕様不備	①では、残数表示インジケータ2の時間枠残数情報配信を30秒周期で要求となっていますが、②では、タイミング:1秒毎と記述されており、仕様書間での不一致があります。正しい時間枠残数情報の要求タイミングについて教えてください。
16	①02_だんだん動物園入場システム概要書.pdf ②13_園内チケットシステム発券機画面仕様書.pdf	①P7 購入 ②S-002-01 いますぐ入場券購入	質問	概要書では「おとなどこどもを合わせて、1回の購入で最大9枚までを購入可能とする。」と記述がありますが、発券機で最大枚数以上を購入しようとした場合、どのような挙動になるでしょうか。
18	13_園内チケットシステム発券機画面仕様書.pdf	S-003-04 時間指定入場券情報確認(入場券確保失敗 残数有)	誤記	画像が2枚貼付けされており、1枚目はいますぐ入場券のメッセージ文言であるよう思われます。見直しをお願いできますでしょうか。
19	①13_園内チケットシステム発券機画面仕様書.pdf ②21_Webチケットシステム画面仕様書.pdf	-	質問	支払い遷移時の文言が発券機では「支払いにすむ」Webチケットシステムでは「お支払い方法選択へ」となっていますが、発券機でも「お支払いにすむ」もしくはWebチケットシステムに合わせ「お支払い方法選択へ」に統一してあげたほうがユーザーとして統一されているように見え、良いと感じましたがいかがでしょうか。
20	①02_だんだん動物園入場システム概要書.pdf ②11_園内チケットシステム要求仕様書.pdf	①P7 入場券購入・発券・入場 ②P20 EK-100 入場管理起動	質問	発券機での購入の時間指定入場券の購入可能時間について、「時間指定種別 概要」の図中では、現在時刻の"次の"次の時間枠から購入できるように読み取れましたが、文章中では「現在時刻の次の時間枠から当日最終入場枠までの30分間隔で入場時刻を決めることができる券」となっております。 11_園内チケットシステム要求仕様書.pdfの入場管理で管理を引き継ぐ時間枠の説明として、「現在時刻を含む枠」「その次の枠」および「次々枠」という記載があることから、図の表現が正しく、文章中の表現に齟齬があるように思われます。ご確認をお願いいたします。 テスト設計書としては、図中の表現を正として作成を進めます。

ツール選定におけるアピールポイント

複数人でテスト開発したり、履歴を管理することも想定したツールを選定

テスト設計成果物	Markdownで記述し、GitHubで管理
RDRAモデル	Miroで作成
シーケンス図	mermaid(Markdown)で記述し、GitHubで管理
要求分析モデル	MindMupで作成
テストアーキテクチャ	draw.io + mermaid(Markdown)で作成し、GitHubで管理
テストケース	Google スプレッドシートで作成

テキストデータで扱うため、差分がわかりやすく、複数人での開発時はPRを活用したテスト開発も可能

同期的な編集は不向きであるが、他のマインドマップツールとの互換性や、クラウドストレージでの管理のしやすさを考慮

まとめ

- 多面的な視点でテスト要求を抽出できた
- テスト設計の意図を各成果物間で途切れさせないよう、テスト要求分析からテスト詳細設計までの繋がりを意識した表現方法を目指した
- 生成AIをテスト設計に活用するための具体案を示せた

ご清聴ありがとうございました



テスト設計をされていて面白い！と思える
テストベースを提供くださった実行委員・関係者の皆様
ありがとうございました！