

# チーム紹介・位置づけ

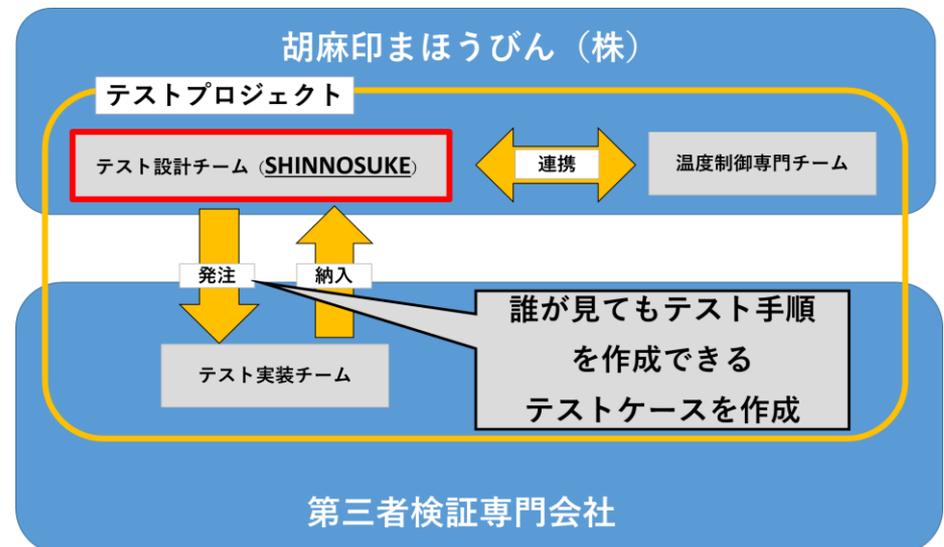
## ・チーム名 SHINNOSUKE

### チームの由来

新しく販売された新潟の「米」のブランド

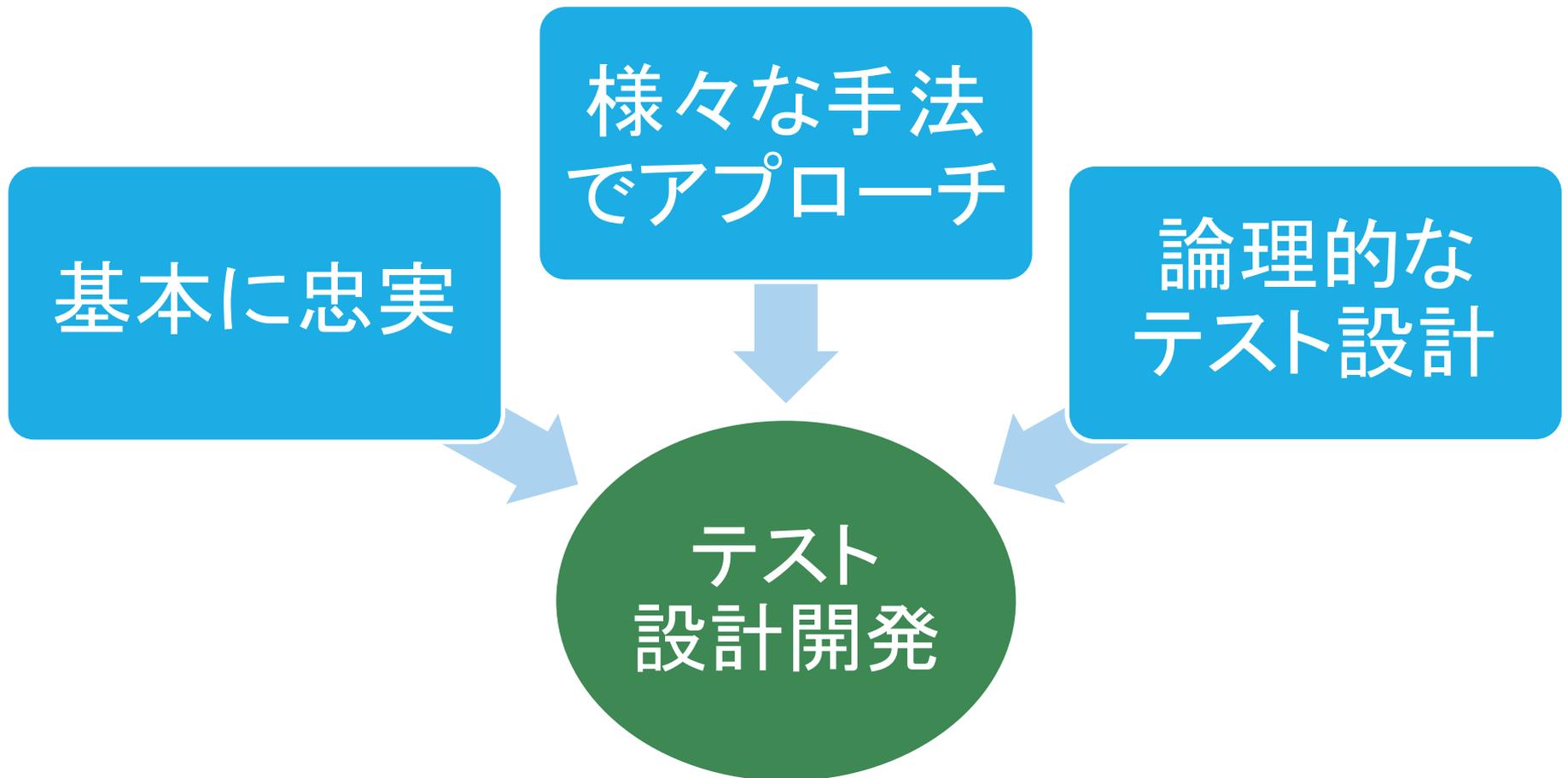
自分たちはまだ、テストを経験したことのない新米であり、これから世の中に出ていくという姿を、自分たちが置かれている状況にちなんで名づけた

## ・チームの位置づけ

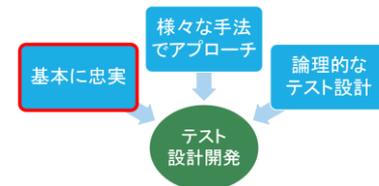


# コンセプト

三つのコンセプトをもとにテスト設計を行う



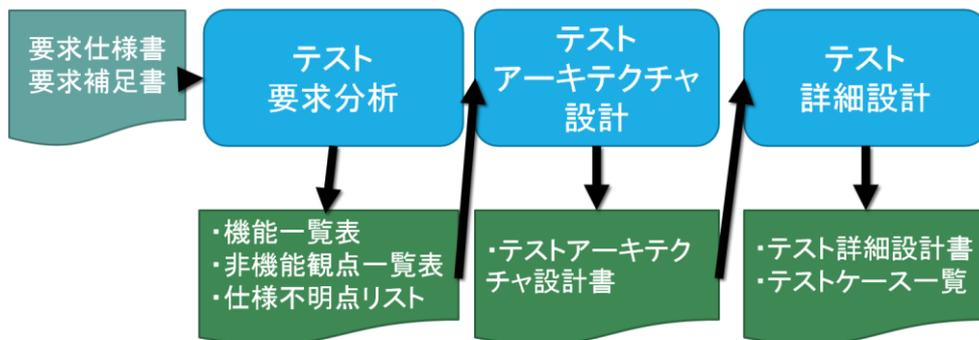
# 基本に忠実



## 1. 代表的なテスト開発プロセス

### 目的

- チュートリアルに従って、テスト開発プロセスを学習する



### 図の説明



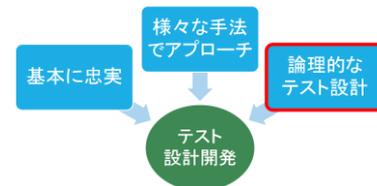
## 2. レビューを行う

### 目的

- 各成果物に対して、抜け漏れ防止



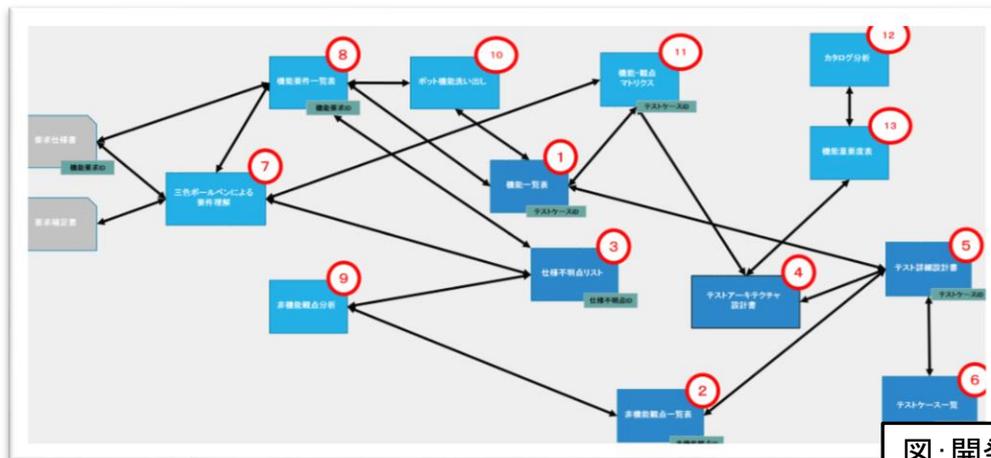
# 論理的なテスト設計



## 1. トレーサビリティを取る

### 目的

- 抜け洩れ防止



図：開発フロー

## 2. 目的を明確化する

### 目的

- ゴールを常に意識する
- 説明を正確に行う

# テスト要求分析(機能洗い出し)



## 1. 三色ボールペン(要求理解)

### 目的

- 要求が何かを明確にする

青: そのまま機能要件に転記できる重要な記載

赤: 仕様不備、不整合などの可能性がある記載

緑: 誤記や不明点、わかりにくい記載や気になった点

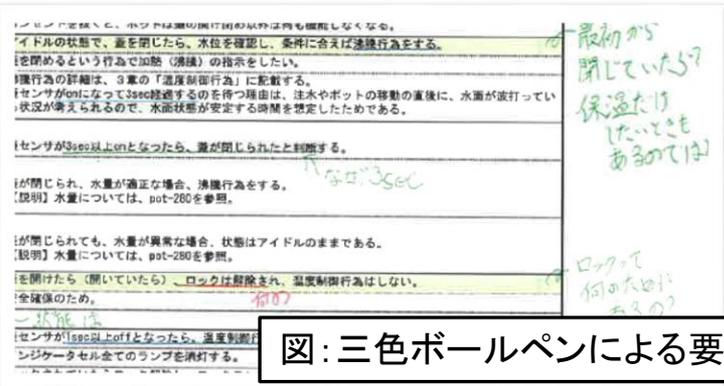


図: 三色ボールペンによる要求理解

## 2. USDM(要件のすりあわせ)

### 目的

- 要求の理由を明確化
- 要求の抜け漏れ防止

内部機能	要求ID	理由	説明
要求 1-6	pot-110	満水センサ	水が許容限界を超えていないかを判断したいから。
	要求 1-6-1	pot-110 - 11	水位が、このポットの許容上限を超えているかどうかを検出する。 水位がポットの許容上限を超えているかどうかを判断したいから。 このセンサがonの時、水位が許容上限を超えていることになる。(フェイルトson)
	pot-110 - 11 - 1	水位がセンサの位置を超えたときは、センサをonにする	フェイルト
要求 1-7	pot-110	満水水位センサ	現状の水位を判断したいから。
	要求 1-7-1	pot-110 - 13	水位を検出する。 現状の水位を判断したいから。 各センサがonの時、その水位より水位が高いことになる。(フェイルトはoff)
	pot-110 - 13 - 1	水位がセンサの位置を超えたときは、センサをonにする。	フェイルト
要求 1-8	pot-110	満水センサ	満水の状態を判断したいから。
	要求 1-8-1	pot-110 - 15	満水が検出しているかを検出する。 満水の状態を判断したいから。 満水が検出しているときonになる。
	pot-110 - 15 - 1	満水が検出しているときは、onになる。	
要求 1-9	pot-110	サーミスタ	ポットの水温を確認したいから。
	要求 1-9-1	pot-110 - 16	ポット内の水温を検出する。 ポット内の水温を確認したいから。 -10℃~15℃
	pot-110 - 16 - 1	ポット内の水温	

図: 機能要件一覧表

## 3. マインドマップ(ポット機能の洗い出し)

### 目的

- USDMから考えられる機能を洗い出す
- 共通の機能を整理する

# テスト要求分析(非機能洗い出し)



## 1.ISO/IEC25010の「利用時の品質特性」

## 2.事件事例分析

### 目的

- 非機能観点を網羅的に考える

### 目的

- 非機能観点をピンポイントで考える

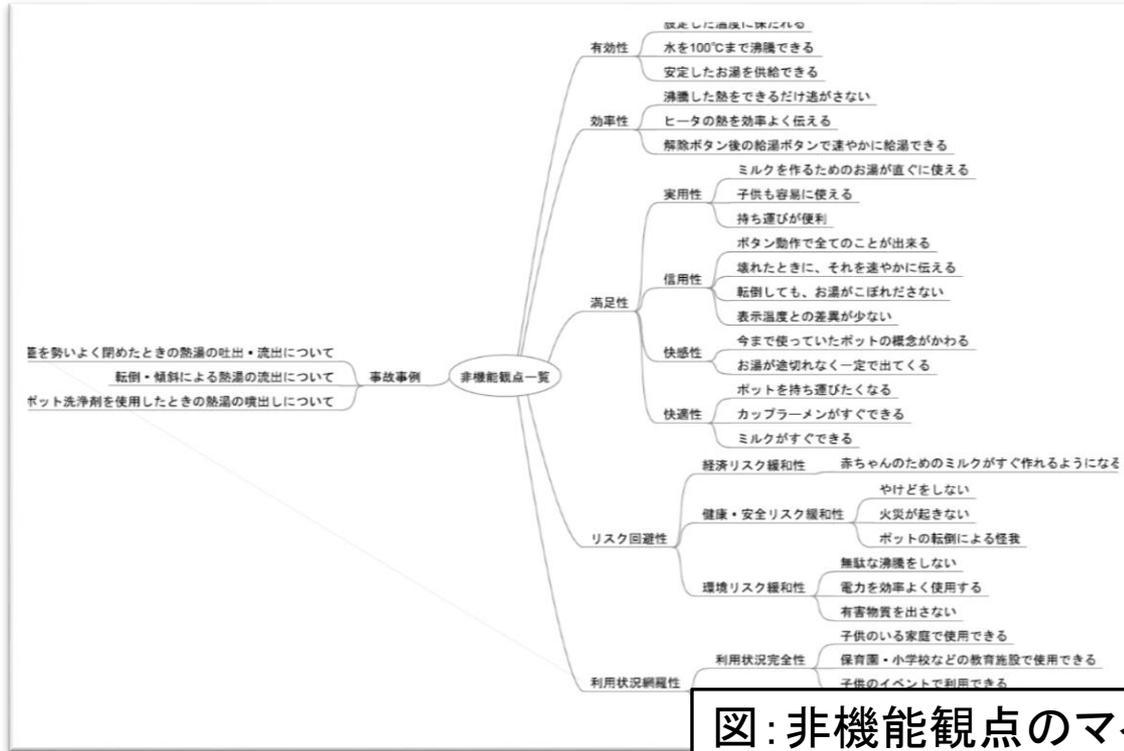


図: 非機能観点のマインドマップ

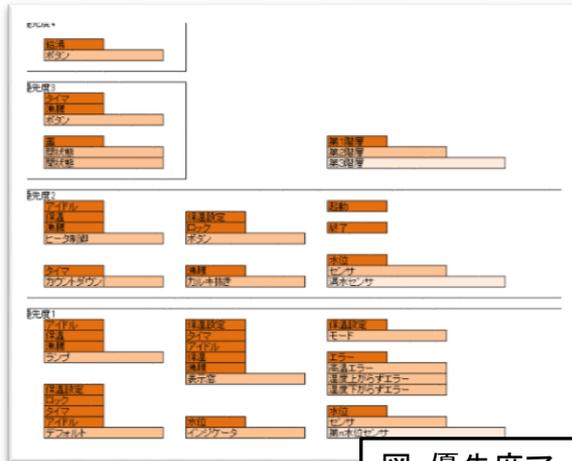
# テストアーキテクチャ設計



## 優先度アーキテクチャ

### 目的

- テスト観点と機能を組みあわせることでテストの優先度を定める

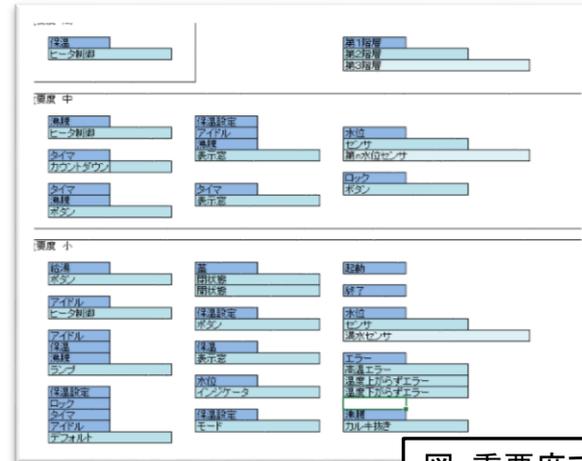


図：優先度アーキテクチャ

## 重要度アーキテクチャ

### 目的

- 一般的な電気ポットが持っている機能を調べることで、テストの重要度を定める



図：重要度アーキテクチャ

### 採用

理由：この方が機能ごとにまとまっていて、テストしやすい

# テスト詳細設計



**優先度4**

**優先度3**

**優先度2**

**優先度1**

**目的**

- テスト条件をなるべく多く網羅する

優先度によるアーキテクチャによって色分け

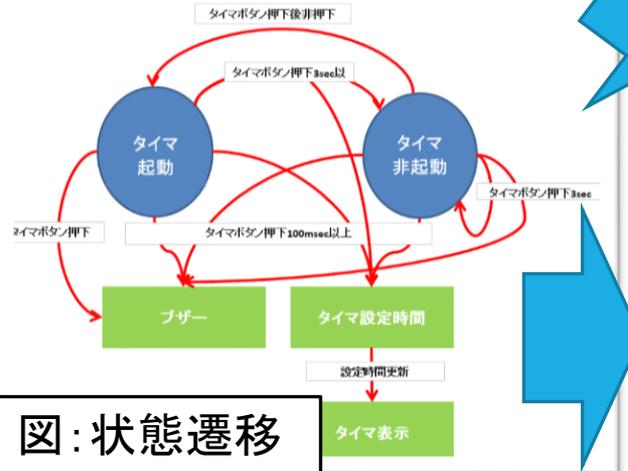


図: 状態遷移

結果	タイマ起動	タイマセット	起動→非起動	非起動→起動
1	T	T	F	F
2	T	T	F	F
3	T	T	F	F
4	F	F	F	F
5	F	F	T	T
6	F	F	T	T
7	F	F	T	T
8	F	F	T	T
9	F	F	T	T
10	F	F	T	T
11	F	F	T	T
12	F	F	T	T

図: デイジジョンテーブル

**目的の方針**

データ情報を入力出力する観点でテストを行う

状態が変化する観点でテストを行う

組合せを行う観点でテストを行う

**要件**

タイマが起動している間にしかかわからず、タイマボタンを100ms以上押される度にタイムアップまでの残り時間を分単位に追加表示し、秒の単位を0secにクリアし、0.1secに設定し、タイマはタイマボタンを押された瞬間から分単位のみで計測し、5秒のタイマ残り時間を表示する。0min0secから最大9min0secまでセットすることができる。60min0secのときに、更にタイマボタンを1回100ms押されると、1min0secをセットしたことになる。1→2→3→...→59→59→60→1→タイマボタンが押れた時、タイムアップまでの残り時間が1分加算される毎に、プザーを50ms押さなければならない。タイマボタンを3sec以上押すとタイマはタイマボタンを100ms押された後、0min0secにリセットされ、タイマが停止する。

図: テスト詳細設計書

ボタン	テストケース	テスト条件	期待する結果	
ボタン	TC06-01-01	・チェック解除中 ・コンソント接続している ・電源モニターなし ・電源停止中	圧電検出で検出ボタンを100ms以上押す	検出が行われる
	TC06-01-02	・チェック解除中 ・コンソント接続している ・電源モニターなし ・電源停止中	圧電検出で検出ボタンを100ms以上押す	検出が行われない
	TC06-01-03	・チェック解除中 ・コンソント接続していない	圧電検出で検出ボタンを100ms以上押す	検出が行われない
	TC06-01-04	・チェック解除中 ・コンソント接続している ・電源モニターあり	圧電検出で検出ボタンを100ms以上押す	検出が行われない
	TC06-01-05	・チェック解除中 ・コンソント接続している ・電源モニターなし ・電源停止中	圧電検出で検出ボタンを100ms以上押す	検出が行われない
	TC06-01-06	・チェック解除中 ・コンソント接続している ・電源モニターなし ・アイドル状態	圧電検出で検出ボタンを100ms以上押す	検出が行われない
カウントダウン	TC07-01-01	・タイマ非起動	圧電検出でタイマボタンを100ms以上押して終了	1分後にカウントダウンが開始する 1分経過にプザーを100ms押せば0分経過 タイマボタンが無効(100ms以上押しても何も起こらない)となる セット済みのタイマ値のカウントダウンは
	TC07-01-02	・タイマ起動中 ・コンソント接続している ・電源モニターなし	圧電検出でタイマボタンを100ms以上押す	タイマがリセットされる
	TC07-02-02	・タイマ非起動	圧電検出でタイマボタンを100ms以上押す	タイマがリセットされる

図: テストケース一覧