Agile Japan 2016

あなたとつくるアジャイル

青木 智英 氏
株式会社コードクオリティ

2003年アジャイルプロセス協議会発足時からの会員で当時からアジャイルを実践。「こうあるべき」という開発を目指すのではなく、現実の予算・期間・リソースの中での最適解を見つけることに力を入れる。

AgileJapan2015公認リポーター。

高柳 謙 氏
ダイアログデザイン

企業向け研修コンサルタント兼ファシリテーター。ファシリテーターについては主に企業外の活動で活動を行っていたが、2012年からファシリテーションを用いたチーム研修を企業内で実施。

「アジャイルの魂（AgileJapan2015）」寄稿
Agile Japan 2011クロージング・ワールド・カフェ
Agile Japan 2012 Tokyo・ワールド・カフェ

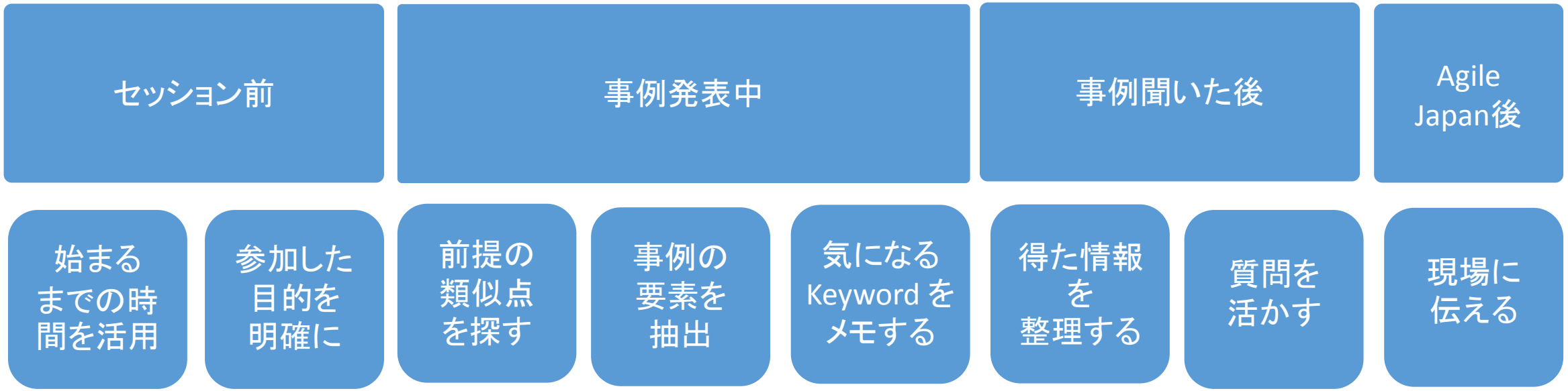
D-1：公募セッション（チーム/技法/成長）

請負で企画・開発・運用・拡張まで担当するアジャイルチーム

【前説】

D-1：セッション「請負で企画・開発・運用・拡張まで担当するアジャイルチーム」の楽しみ方

ガオ流事例発表の聞き方講座



【事例の前提】

会社紹介

青木 智英 氏
株式会社コードクオリティ

2002年1月 「株式会社ジェイトランス」設立

大学院在籍中、数々の炎上プロジェクトにプログラマとして参加

あるプロジェクトのメンバで会社設立（と同時に大学院中退）

2013年10月 「株式会社コードクオリティ」に社名変更

ソースコード重視のスタイルを確立するためにリスタート

社員数12名、プロジェクトは初回リリースまで3～6ヶ月のものが多い

【事例の前提】

事前知識

■ システム開発における契約の種類

種類	作業場所	指示内容	完成義務
派遣	客先	作業の直接指示	なし
準委任	客先 or 自社	業務内容のみ	なし
請負	客先 or 自社	業務内容のみ	あり

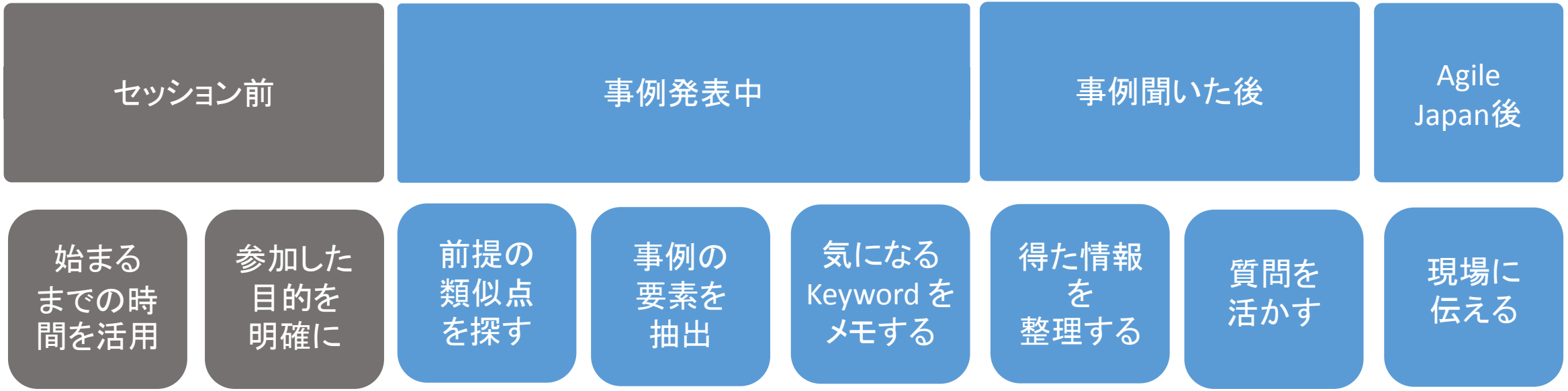
■ コードクオリティの仕事の請け方

・CQは客先常駐しないので、準委任（自社）と請負（自社）で行っている。

【セッショッン前】

D-1：セッション「請負で企画・開発・運用・拡張まで担当するアジャイルチーム」の楽しみ方

ガオ流事例発表の聞き方講座



プロセス

契約

スキルマップ

【事例のきっかけ】

Q:なぜ、プロトタイプ開発を始めたのか？

【事例のきっかけ】

Q:なぜ、プロトタイプ開発を始めたのか？

■ 開発の基本はウォーターフォール！

「いつできるかわからない」「いくらかかるかわからない」では依頼できない

「リリース日」と「予算」を決めてゴールを目指す

【事例のきっかけ】

Q:なぜ、プロトタイプ開発を始めたのか？

■ 開発の基本はウォーターフォール！

「いつできるかわからない」「いくらかかるかわからない」では依頼できない

「リリース日」と「予算」を決めてゴールを目指す

■ しかし、ウォーターフォールには問題点がある

- 仕様変更は避けられない
  - 使ってみての感想は使ってみないと得られない
    （想像と実物は違う）
  - 世の中はどんどん変化するので要件も変化するのが当然
    （今日良いことが明日良いとは限らない）

- 見積もりが困難（仮に途中で仕様変更は発生しないとしても）
  - 作業項目の洗い出しが要件定義や設計作業を行っている状態に陥る
    （マスタメンテは一括登録？エラーチェックはどこまで必要？）
  - 新しい技術や新しい環境（OSや端末）は試してみないとわからない
    （ロジック部分は合っていても表示関連は崩れることが多い）

【プロトタイプ開発】

Q:どうやってプロトタイプ開発を進めているのか？

【プロトタイプ開発】

Q:どうやってプロトタイプ開発を進めているのか？

■ ユーザーが触ってみれるものをまず開発する。（モックアップ＝プロトタイプと呼ぶ）

プロトタイプ：全体の流れや使い勝手を確認するためのもの（正常系をメインに異常系は確認に必要なだけ）

- 正常系：通常の操作で動作するもの（異常な操作を想定しない）
- 異常系：入力ミスや不正アクセスでもシステムが稼働し続けるのも

【プロトタイプ開発】

Q:どうやってプロトタイプ開発を進めているのか？

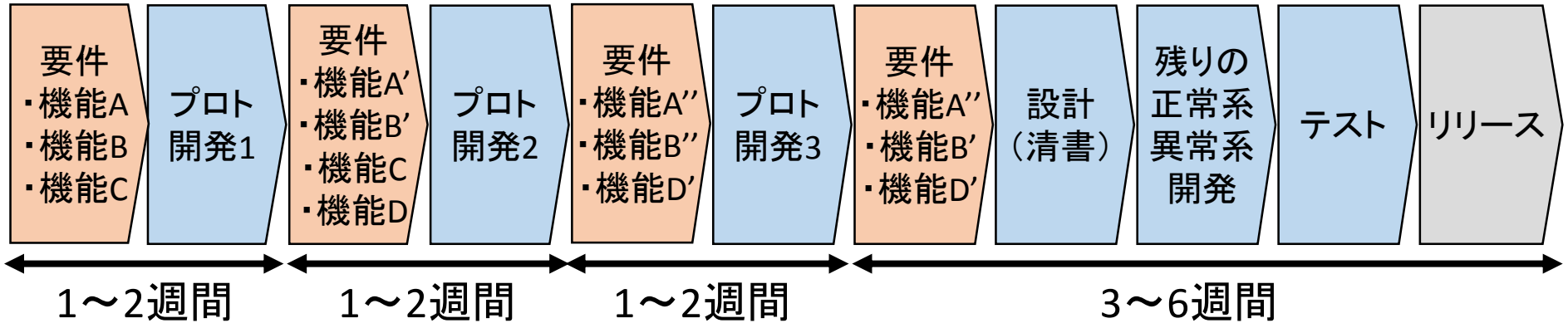
■ ユーザーが触ってみれるものをまず開発する。（モックアップ＝プロトタイプと呼ぶ）

プロトタイプ：全体の流れや使い勝手を確認するためのもの（正常系をメインに異常系は確認に必要なだけ）

・正常系：通常の操作で動作するもの（異常な操作を想定しない）

・異常系：入力ミスや不正アクセスでもシステムが稼働し続けるのも

■ 最終的にはウォーターフォールに落とし込む。

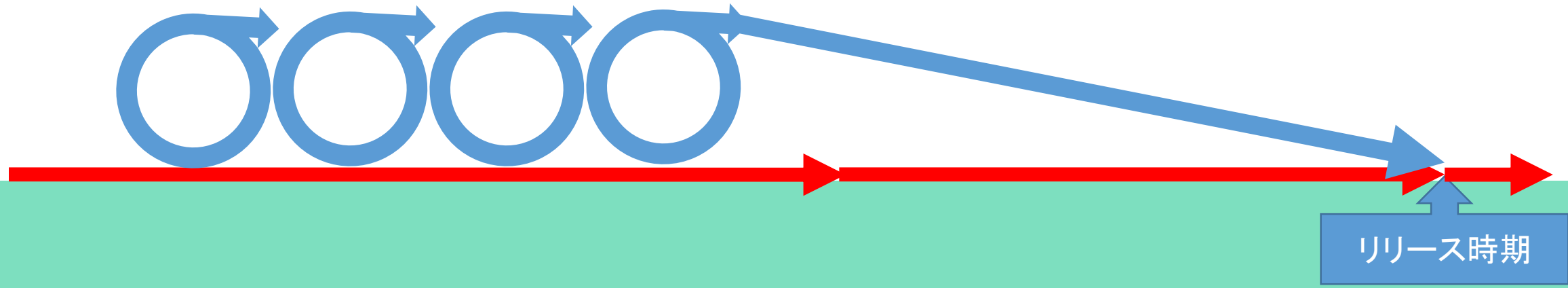


プロセス

契約

スキルマップ

プロトタイプ開発　＋　ウォーターフォール



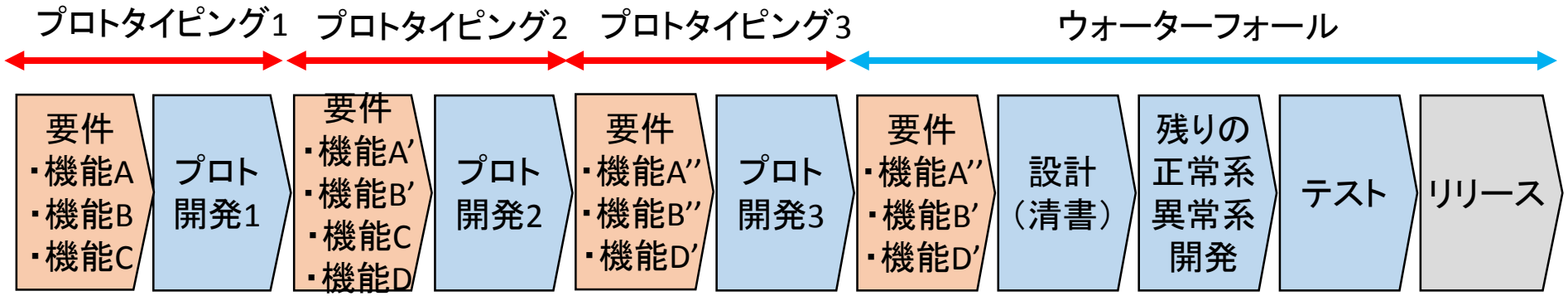
【プロトタイプ開発】

Q:なぜ、プロトタイプ開発でできているのか？

【プロトタイプ開発】

Q:なぜ、プロトタイプ開発でできているのか？

■ プロトタイプ開発とウォーターフォールのハイブリッド



ウォーターフォール部分の見積もり精度

高い

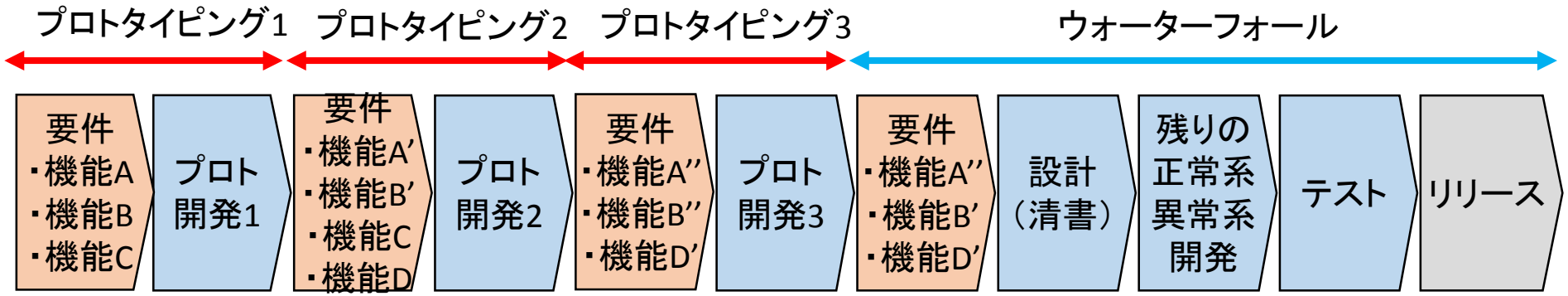
ウォーターフォール前にコードの一部ができている

ウーオーフォールの期間が短縮するのでぶれない

【プロトタイプ開発】

Q:なぜ、プロトタイプ開発でできているのか？

■ プロトタイプ開発とウォーターフォールのハイブリッド



ウォーターフォール部分の見積もり精度

高い

■ 契約の形

契約	完成義務	瑕疵	見積もり
準委任	なし	なし	人数単位（プロトの内容によって都度変更）
請負	あり	あり	機能単位

ウォーターフォール前にコードの一部ができている

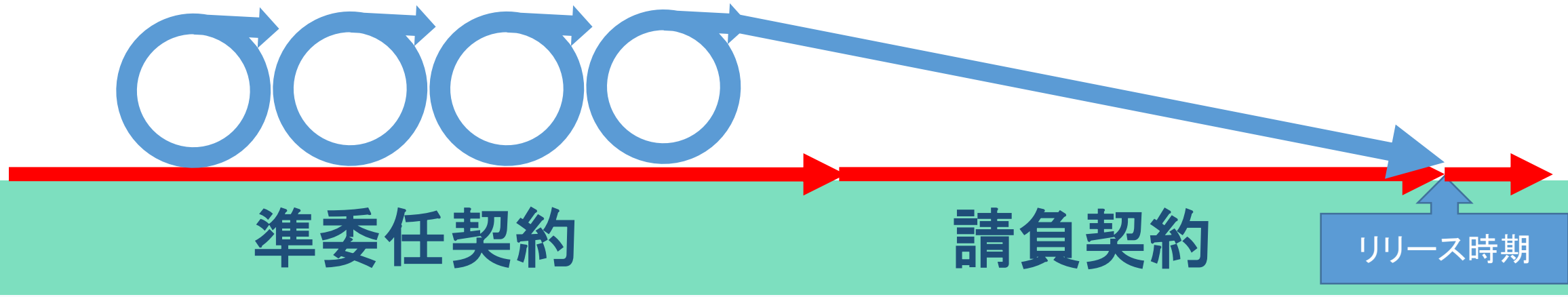
ウーオーフォールの期間が短縮するのでぶれない

プロセス

契約

スキルマップ

プロトタイプ開発　＋　ウォーターフォール



【プロトタイプ開発】

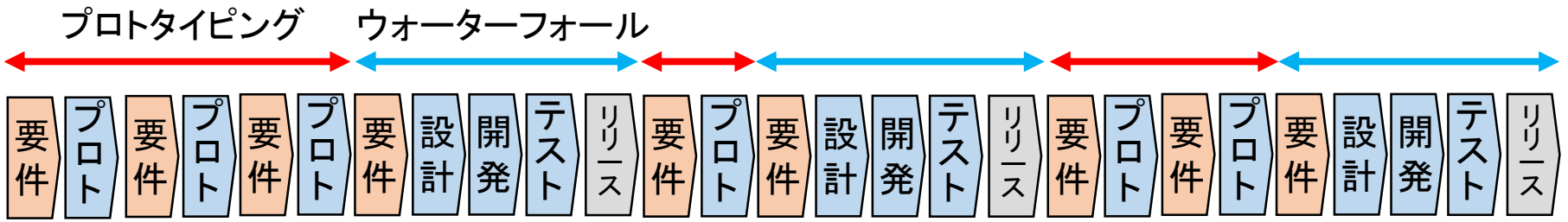
Q:プロトタイプ開発で注意していることは？

【プロトタイプ開発】

Q:プロトタイプ開発で注意していることは？

■その１:システムの拡張における品質

システムはリリースしたら終わりではなく要求に応じて拡張していく必要がある



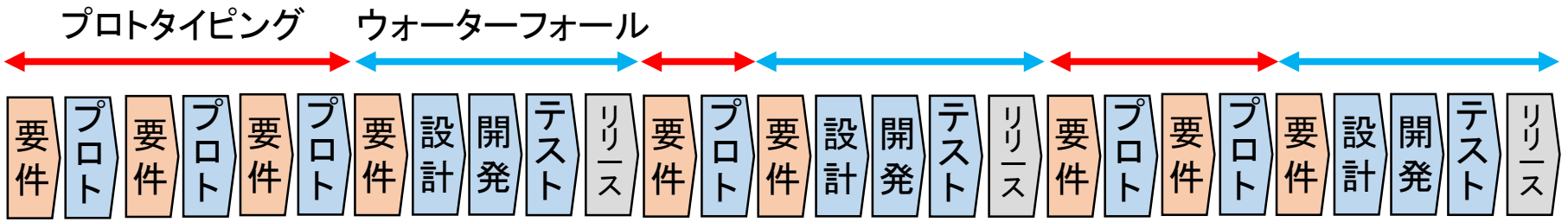
機能追加に多くの費用や時間がかかるとしたらシステムの品質は低いと言える

【プロトタイプ開発】

Q:プロトタイプ開発で注意していることは？

■その１:システムの拡張における品質

システムはリリースしたら終わりではなく要求に応じて拡張していく必要がある



機能追加に多くの費用や時間がかかるとしたらシステムの品質は低いと言える

テストは動作保証しているだけで品質保証にならない

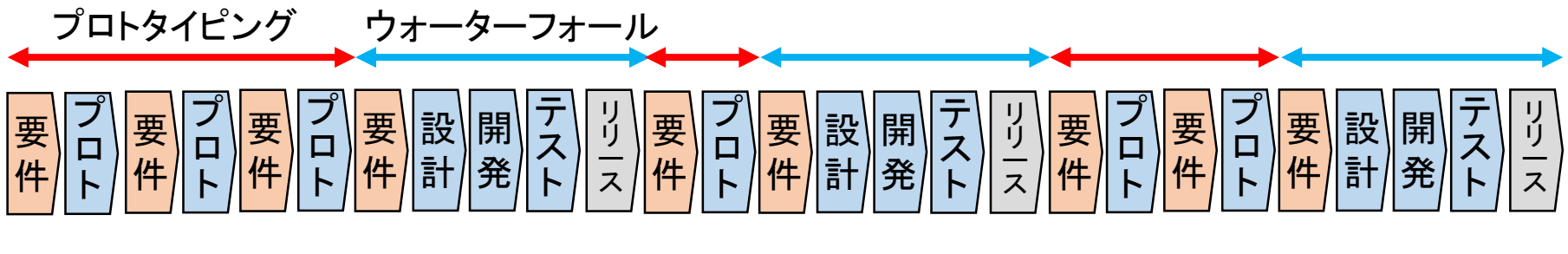
拡張しやすいソースコードを維持することがシステム全体の品質を高める

【プロトタイプ開発】

Q:プロトタイプ開発で注意していることは？

■その２:価値のあるシステムを開発すること

予算と時間をかけて
不具合のないものを
リリースする

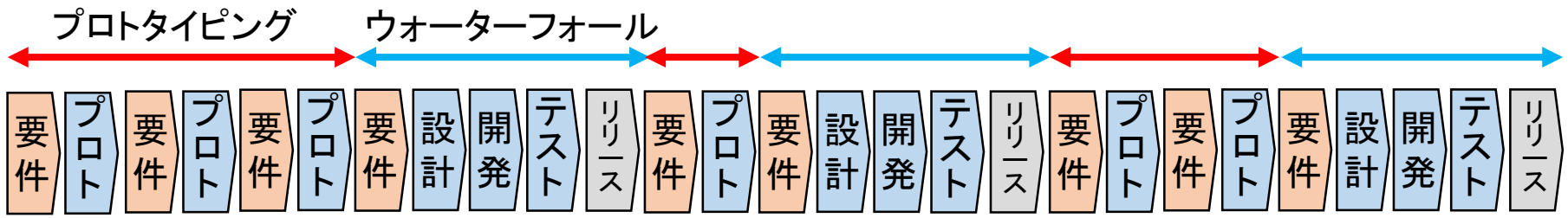


【プロトタイプ開発】

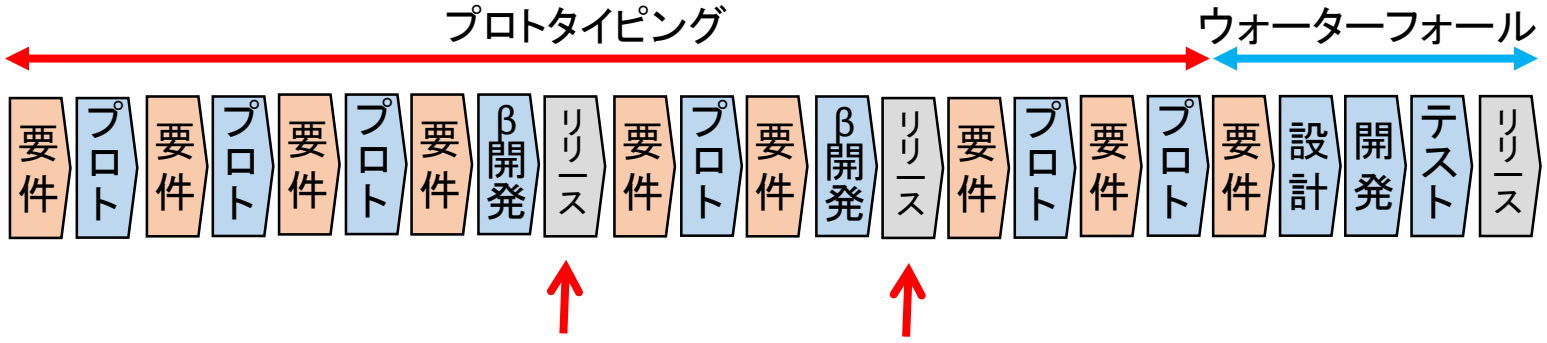
Q:プロトタイプ開発で注意していることは？

■その2:価値のあるシステムを開発すること

予算と時間をかけて
不具合のないものを
リリースする



不具合があっても
予算と時間をかけずに
リリースする



不具合を残したままリリース　（永遠のβ版もアリ）

※ネット環境は修正版の配布が容易なので「不具合なく動作すること」と同様に「早く便利なサービスを提供する」ことにも価値がある

ここだけの話

プロトタイプ開発をお客さんと始めるとは？

1　プロトタイプ開発を繰り返すことで実際に動くものに触れながら
　　要件を確定するためには？

2　プロトタイプとウォーターフォールで契約を分けるには？

3　予算や時間は限りがあるものなので「不具合をなくす」こと
　　「早く便利なサービスを提供する」ことのバランスを考える

ここだけの話

プロトタイプ開発をお客さんと始めるとは？

1　プロトタイプ開発を繰り返すことで実際に動くものに触れながら
　　要件を確定するためには？

2　プロトタイプとウォーターフォールで契約を分けるには？

3　予算や時間は限りがあるものなので「不具合をなくす」こと
　　「早く便利なサービスを提供する」ことのバランスを考える

ここだけの話

プロトタイプ開発をお客さんと始めるとは？

1　プロトタイプ開発を繰り返すことで実際に動くものに触れながら
　　要件を確定するためには？

2　プロトタイプとウォーターフォールで契約を分けるには？

3　予算や時間は限りがあるものなので「不具合をなくす」こと
　　「早く便利なサービスを提供する」ことのバランスを考える

ここだけの話

プロトタイプ開発をお客さんと始めるとは？

1　プロトタイプ開発を繰り返すことで実際に動くものに触れながら
　　要件を確定するためには？

2　プロトタイプとウォーターフォールで契約を分けるには？

3　予算や時間は限りがあるものなので「不具合をなくす」こと
　　「早く便利なサービスを提供する」ことのバランスを考える

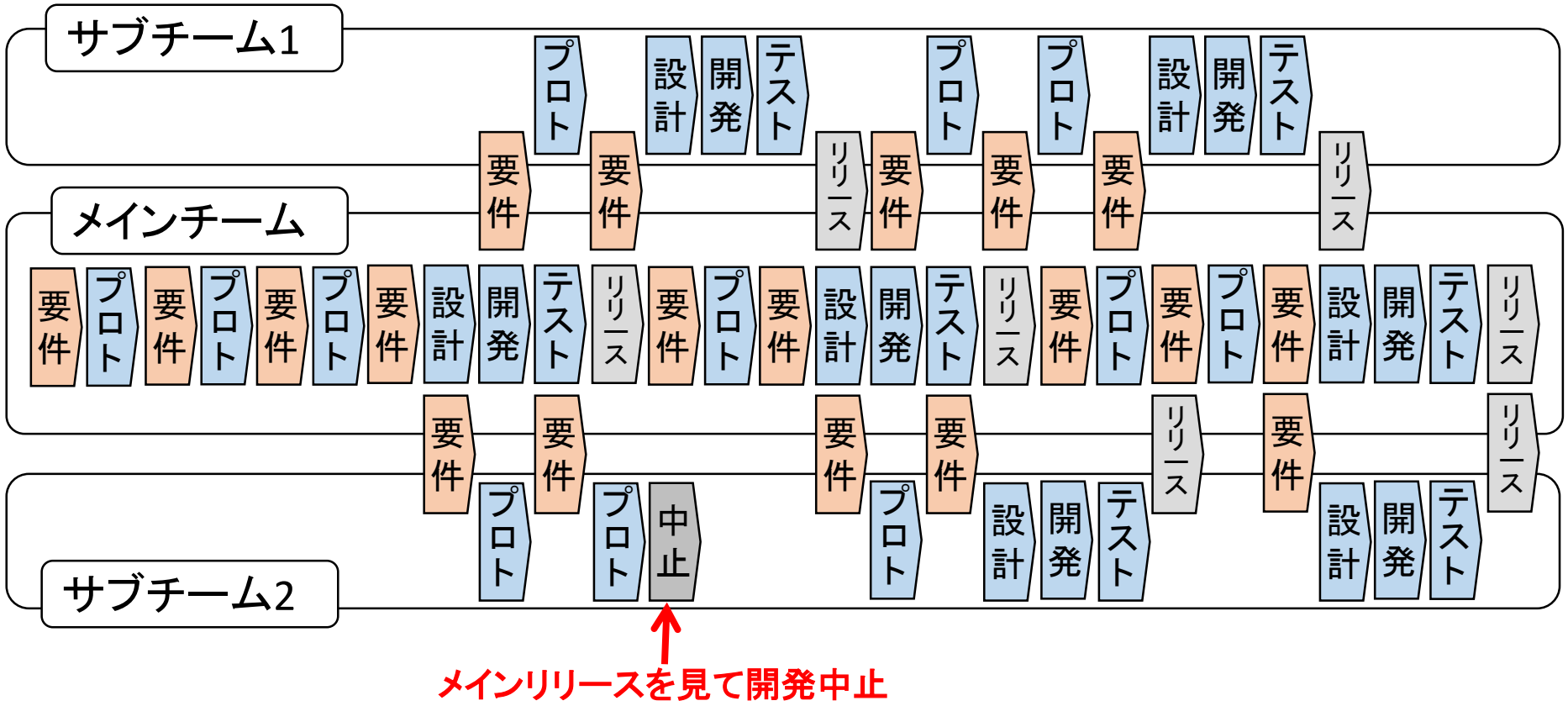
【プロトタイプ開発を支える体制】

Q：開発のピークが偏ると思いますが、どうしていますか？

【プロトタイプ開発を支える体制】

Q：開発のピークが偏ると思いますが、どうしていますか？

■ 1つのプロジェクト内でのチーム編成



【プロトタイプ開発を支える体制】

Q:開発チームはどのように作っていますか？

【プロトタイプ開発を支える体制】

Q：開発チームはどのように作っていますか？

■ チームのスキルマップを作成してチーム状態を把握しています。

【プロトタイプ開発を支える体制】

Q:開発チームはどのように作っていますか？

■ チームのスキルマップを作成してチーム状態を把握しています。

	交渉者	管理者	応援者	アーキテクト	プログラマ	育成者	UIデザイナ	インフラ	社内基盤
Level.1	★	★	★	★	★	★	★	★	★
Level.2	☆	☆	☆	☆	☆	☆	☆	☆	☆
Level.3	☆	☆	☆	☆	☆	☆	☆	☆	
Level.4	☆	☆		☆	☆				
Level.5				☆	☆				

【プロトタイプ開発を支える体制】

Q:開発チームはどのように作っていますか？

■ チームのスキルマップを作成してチーム状態を把握しています。

交渉者	管理者	応援者	アーキテクト	プログラマ	育成者	UIデザイナ	インフラ	社内基盤
交渉	進捗管理	社内向けコミュニケーション	設計	プログラミング	育成	UIデザイン	インフラ	社内ツール
傾聴	予算管理	宴会部長	アーキテクト	テスト				

【プロトタイプ開発を支える体制】

Q:開発チームはどのように作っていますか？

■ チームのスキルマップを作成してチーム状態を把握しています。

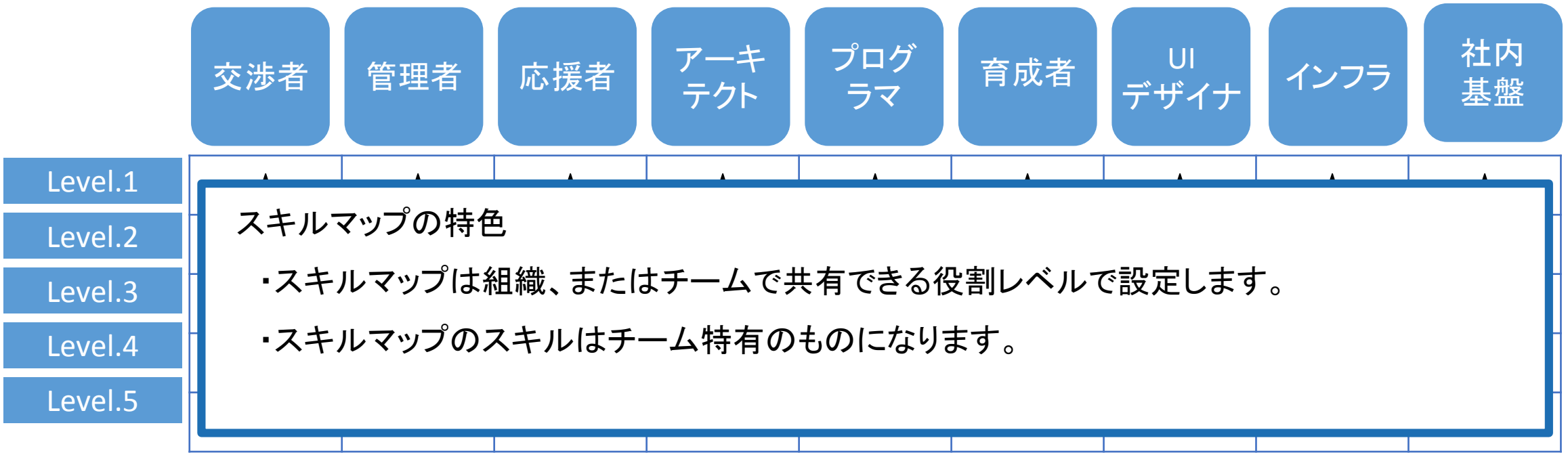
	交渉者	管理者	応援者	アーキテクト	プログラマ	育成者	UIデザイナ	インフラ	社内基盤
Level.1						★	★	★	★
Level.2						☆	☆	☆	☆
Level.3						☆	☆	☆	
Level.4									
Level.5									

Level1	新人
Level2	一般
Level3	サブリーダーできる（決定権がない）
Level4	メインリーダーできる（決裁権を渡せる）
Level5	マスターレベル（CXOレベル）

【プロトタイプ開発を支える体制】

Q:開発チームはどのように作っていますか？

■ チームのスキルマップを作成してチーム状態を把握しています。



【プロトタイプ開発を支える体制】

Q:メインチームとサブチームについて詳しくおしえてください。

■ メインチームとサブチームで必要なスキルレベルが違っています。

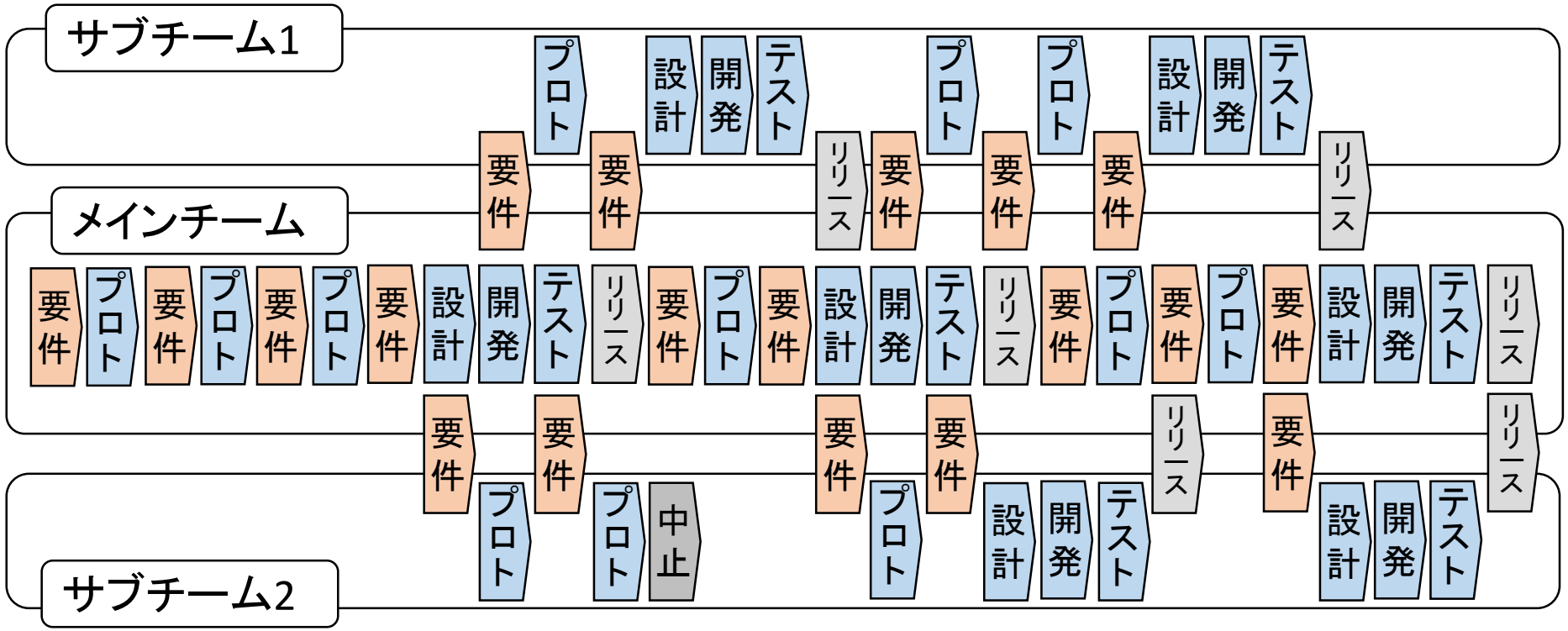
	メインチーム								
	交渉者	管理者	応援者	アーキテクト	プログラム	育成者	UI デザイナ	インフラ	社内 基盤
Level.1	☆	☆	☆	☆	☆	☆	☆	☆	☆
Level.2	☆	☆	☆	☆	☆	☆	☆	☆	☆
Level.3	☆	☆	☆	☆	☆	☆	☆	☆	
Level.4	☆	☆		☆	☆				
Level.5				☆	☆				

	サブチーム								
	交渉者	管理者	応援者	アーキテクト	プログラム	育成者	UI デザイナ	インフラ	社内 基盤
Level.1	☆	☆	☆	☆	☆	☆	☆	☆	☆
Level.2	☆	☆	☆	☆	☆	☆	☆	☆	☆
Level.3	☆	☆			☆	☆			

【プロトタイプ開発を支える体制】

Q:メインチームとサブチームについて詳しくおしえてください。

■ メインチームとサブチームで必要なスキルレベルが違っています。

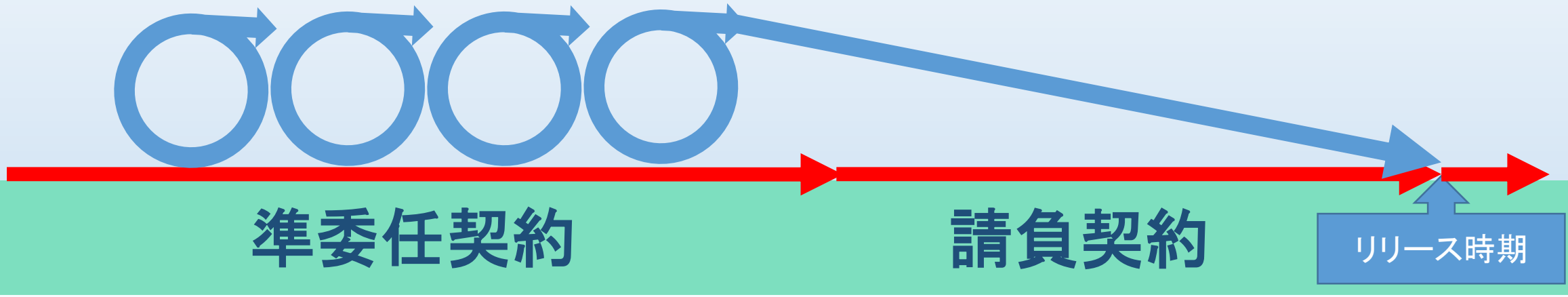


プロセス

契約

スキルマップ

プロトタイプ開発　＋　ウォーターフォール



	交渉者	管理者	応援者	アーキテクト	プログラマ	育成者	UI デザイナ	インフラ	社内基盤
Level.1									
Level.2									
Level.3									
Level.4									
Level.5									

【事例聞いた後】

D-1：セッション「請負で企画・開発・運用・拡張まで担当するアジャイルチーム」の楽しみ方

ガオ流事例発表の聞き方講座

【Agile Japan後】

D-1：セッション「請負で企画・開発・運用・拡張まで担当するアジャイルチーム」の楽しみ方

ガオ流事例発表の聞き方講座