特集号
投稿論文

社内でアワビを飼育して実践するアジャイル開発―クラウドサービス：NEC養殖管理ポータルの開発―

西潟 憲策[†1]　佐々木 裕之[†1]　安藤 寿之[†1]　野沢 善浩[†1]　中谷 貴子[†1]

[†1] NECソリューションイノベータ（株）

アジャイル開発は，開発プロセスだけでなく，プロジェクト管理の考え方も従来の方法とは大きく異なる．ベンダとして，さまざまなプロジェクトにおいてアジャイル開発を適用し，成功に導くためには，アジャイル開発の経験と実績を積み重ねることが重要である．筆者らは，これまでにもNECグループ内のツール開発やクラウドサービス開発において，積極的にアジャイル開発を採用し，ノウハウを蓄積するようにしてきた．今回，当社の新規ビジネス領域のクラウドサービス開発において，アジャイル開発を適用した．本稿では，当事例について，要件の検討と変更への適用の取り組みを中心に紹介する．

1. はじめに

要件の変化や新たな要件の発生にも柔軟に対応できる開発として，アジャイル開発が注目を浴びている．ウォーターフォール開発は，はじめにすべての要件を明確にしてから開発を進め，開発のすべてが完了してから価値を提供する．一方で，アジャイル開発は，優先順位の高い要件から開発を進め，価値を順次提供していく．

近年は日本のベンダもアジャイル開発への取り組みを活発化させつつある．最近は，エンタープライズアジャイルという言葉も登場し，さまざまな文脈でそれぞれの定義において使用されている．本稿では，エンタープライズアジャイルという言葉は，アジャイル開発やハイブリッドアジャイルなどの開発プロセスを指す言葉としてではなく，従来の企業内プロセスにアジャイル開発を適用するためのルール整備や，人材育成までも含めた，企業におけるアジャイル開発への取り組み全般を指す言葉として使用する．

NECグループでは，NECソフトウェアグループにおける現場の草の根活動として2007年にアジャイルコミュニティが発足した．当コミュニティは，アジャイル開発についての情報交換や定期的な勉強会を実施しており，現在に至るまで活動を継続している．また，2010年と2011年にかけてはアジャイルタスクフォースが発足し，エンタープライズアジャイルを進める上での課題を検討した．これらの活動を経て，2012年にNECグループにおけるアジャイル開発の道標としてアジャイル開発ガイドを発行し，NECアジャイルとした．NECグループでは，ウォーターフォールモデル向けの品質管理技法として，ソフトウェア品質会計という独自の技法を考案し，開発するソフトウェアの品質確保に長年取り組んできた実績がある．NECアジャイルは，この品質会計で得た品質確保の知見をアジャイル開発に応用している．また，プロセスとマネジメントのフレームワークであるScrumをベースとし，eXtreme Programming（以下XPと記す）のエンジニアリングのプラクティスの中から，特に品質向上に効果が見込めるプラクティスを取り入れている．NECアジャイルについては，すでに誉田が報告しているので参照いただきたい[1].

NECアジャイル
- **品質会計から得た知見の応用と適用**
- **Scrumによるプロセスとマネジメント**
- **XPによる品質向上プラクティス**

アジャイル開発は，ともすると，単なる開発のプロセスや手法として捉えられてしまうが，根底にあるプロジェクト管理の考え方も従来の方法とは異なる．ウォーターフォール開発は計画駆動型であり，要件を定め，リソースと期間を計画して開発する．これに対し，アジャイル開発は実測駆動型であり，リソースと期間を定め，優先順位の高い要件から，まずは実際に開発していく．そして，この生産性を計測し，これに基づいて要件範囲を調整して計画を見直す．アジャイル開発では，このように，これまで前提としてきたプロジェクト管理の考え方と方法が従来とは大きく異なる．よって，今後，アジャイル開発をうまく活用していくためには，組織として

多くの経験と実績を積み重ねてノウハウを蓄積することが重要である．

しかし，一方で，自社開発が中心の欧米とは異なり，ユーザ・ベンダ型の開発が中心である日本の産業構造下においては，アジャイル開発を適用するには，顧客との契約をはじめとするさまざまな課題が多く，アジャイル開発の経験と実績を積む機会が少ない．そこで，筆者らはNECグループ内のツール開発やクラウドサービス開発において，積極的にアジャイル開発を採用してノウハウを蓄積し，エンタープライズアジャイルを進めている．本稿では，当社の新規ビジネス領域のクラウドサービス開発においてアジャイル開発を適用した事例について紹介する．第2章でプロジェクトの概要を述べ，第3章で要件の検討および変更への適応のための取り組みを具体的に紹介する．

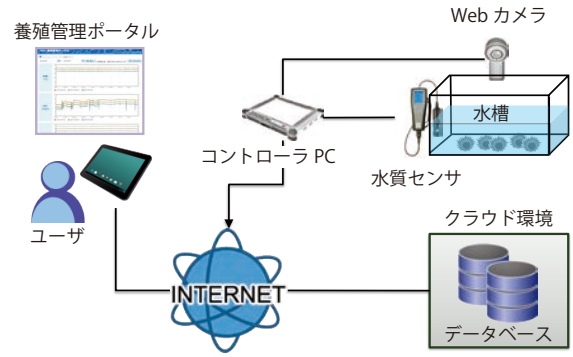
自主プロジェクトに積極的に
アジャイル開発を適用し，ノウハウを蓄積

2. プロジェクトの概要

2.1 背景と対象システム

NECグループは安全，安心，効率，公平な社会価値を創造する社会ソリューション事業に注力しており，その一環として始めた新規事業創出プロジェクトにおいて水産養殖業を支援するビジネスモデルが選ばれた．水産養殖事業者や研究機関をはじめとする業界関係者にヒアリングを重ね，調査を進める中で，食の安心・安全・安定供給を事業の目的とし，水産養殖の環境と作業の可視化，および養殖ノウハウの蓄積を実現する仕組みを構築することとした．

当プロジェクトで開発したNEC養殖管理ポータル（以下，養殖管理ポータルと記す）は，水槽やいけすに設置したセンサが気温，水温，溶存酸素濃度，塩分，pHなどの水質を測定し，水質情報をクラウド上のサーバに定期的に送信する（図1）．ユーザはあらかじめ許容可能な水質情報の範囲をシステムに登録しておき，センサで計測された値がこれを超えるとサーバはアラートメールを送信する．これにより，ユーザは水質の異常発生をリアルタイムに知ることができる．早急に必要な対応が取れるため，養殖対象の死滅を防ぐことができる．また，ライブカメラを使用して飼育状況を遠隔地からモニタリングすることができる．さらに，養殖管理ポータルでは，



（a）システム概要

ダッシュボード画面

日誌画面　　センサグラフ画面

（b）画面イメージ

図1　NEC 養殖管理ポータル

センサから取得する水質情報に加えて，餌の種類や量，成長状況，投薬や掃除の状況，問題の発生状況などを日誌として日々記録することができるため，養殖ノウハウを蓄積して養殖の改善を図ることができる[2].

NEC 養殖管理ポータルにより
養殖環境の異常をリアルタイムに検知，
養殖農法を蓄積して改善

2.2 プロジェクトの期間と体制

本開発プロジェクトの期間は，2013年7月から2015年3月までの1年9カ月間で，後半は運用保守と販促活動が中心であった．体制としては，プロダクトオーナー，開発・運用保守チーム，スクラムマスターに加え，水産養殖アドバイザ，ユーザビリティ専門家，サービスデスクのメンバが参加した（図2）．

水産養殖アドバイザと
ユーザビリティ専門家が参加

開発・運用保守チームは，時期により2～5名の範囲で増減したが，平均3名程度であった．Scrumでは1チームの人数は9名以内を推奨しており，本プロジェクトの開発・運用保守チームはこの範囲内であるため，1チームの体制とした．複数チームにて開発する場合は，機能単位でチームを編成し，機能間のインタフェースを定義した上で，チーム間のコミュニケーションの場を定期的に設ける必要があるが，チーム内のプロセスと実施プラクティスについては本稿に記載する内容を基本的に適用できると考える．

なお，ウォーターフォール開発では，プロジェクト初期の要件定義や設計の段階までは少数メンバで実施し，その後，実装するメンバを一気に増員することが多い．一方で，アジャイル開発では，開発メンバを基本的には一定に維持し，要求分析や要件定義からテストまでの一通りの開発プロセスを常に実施して継続的に開発する．このため，適用する開発プロセスにあわせて，開発するメンバのスキルと人数などのリソース投入計画を検討する必要がある．

本プロジェクトでは，サービスデスクのメンバは顧客からの問合せ等に対応するだけでなく，実際に社内でアワビの飼育を試行しながら開発システムを試用して継続的に評価した．これについては，後述する．また，ユーザビリティ専門家によるユーザビリティ向上の取り組みについては，すでに柳澤らが報告しているので参照いただきたい[3]．

2.3　開発プロセスと実施プラクティス

開発プロセスはNECアジャイルに従い[1]，Scrumのフレームワークを適用して，スプリント期間を2週間とした（図3）[4]．毎週水曜日に定例会を実施し，スプリントレビュー／振り返り／次スプリントのスプリント計画と，要求ワークショップを隔週で交互に開催した．要求ワークショップでは，今後開発するストーリーについてプロダクトオーナーと開発・運用保守チームとで要求と要件についての検討を実施した．毎週の定例会とすることで，開発プロセスにリズムを生じさせる効果があった．また，メンバの日程調整と定例会の開催場所の確保をプロジェクト開始の最初に実施できるため，その後のこれらの調整の手間を省くことができた．

プロダクトオーナーと開発・運用保守チームは，所属部門が異なり，居室のある階も離れている．プロダクトオーナーがいつでも開発状況を確認できるようにするために，プロダクトバックログ，タスク，バグはチケット管理ツールのRedmine Backlogsを用いて管理した[5]．タスク管理では，タスクボード（かんばん）とバーンダウンチャートを利用した．また，プロダクトオーナーと開発・運用保守チームは週2回，一緒にデイリースクラム（朝会）を実施し，毎週の定例会以外にもタイムリーに状況を共有した．デイリースクラムでは開発・運用保守チームのメンバが作業状況を話すだけでなく，プロダクトオーナーも同じように作業状況を話すことにより，ビジネスの状況も全員で共有することができた．ビジネスの状況を共有することは，開発・運用保守チームの開発のモチベーションを維持する効果があった．

XPのプラクティスの中からは継続的インテグレーション，ペアワーク，リファクタリング，ソースコードの共同所有を実施した[6]．継続的インテグレーションではビルド，単体テスト，画面操作も含めた機能テストを自動化した．継続的インテグレーションにより常に動くソフトウェアを確認し，テストの自動化により常にデグレートがないことを確認できることによって，開発・運用保守チームだけでなく，プロダクトオーナーも品質に対して安心してプロジェクトを進めることができた．表1に，本プロジェクトにて実施したプラクティスの一覧を示す．

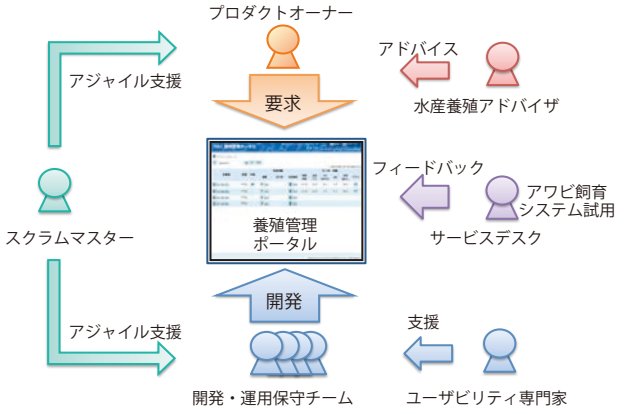


図2　開発の体制

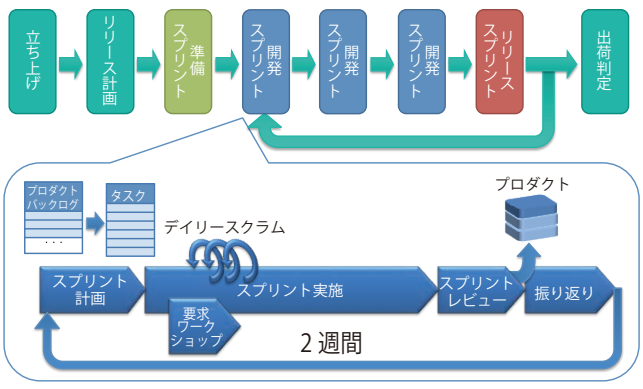


図3　開発プロセス

表 1　実施したプラクティス一覧

インセプションデッキ	リリース計画
プロダクトバックログ	スプリント計画
スプリントレビュー	振り返り
要求ワークショップ	デイリースクラム（朝会）
タスクボード（かんばん）	バーンダウンチャート
継続的インテグレーション	ペアワーク
ソースコードの共同所有	リファクタリング

Scrum のプロセスと XP のプラクティスを適用

2.4　出荷判定と組織へのフィードバック

NECグループでは，製品やシステムをリリースする前に出荷判定というイベントを実施し，リリースして良い品質か否かを組織として出荷判定者が最終判定する．NECアジャイルにおいては，出荷判定項目やアジャイル開発の場合のバグのカウントルールなどを定めている．しかし，アジャイル開発はウォーターフォール開発に比べるとまだまだ事例が少く，アジャイル開発の場合の考え方やその際の出荷判定について現場への理解が進んでいない．そこで，今回，本システムの出荷判定を行うにあたっては，関係者にアジャイル開発の考え方とその際の出荷判定について事前に説明してから出荷判定に臨んだ．出荷判定では，品質メトリクスと品質見解について，①品質の作り込み，②テストの必要十分性，③摘出したバグへの対処の3つの観点に整理して説明を実施し，出荷判定を完了して本システムをリリースした．

NECアジャイルにおいて，アジャイル開発の品質管理について定めているものの，これまで当社では，①アジャイル開発の際のオペレーションプロセスの確立，②アジャイル開発用の出荷判定フォーマットなどの帳票類の整備は進んでいないのが実情であった．そこで，これまで実施してきたNECグループ内のアジャイル開発事例と本事例とをあわせて，現在，上述の①，②を整備し，エンタープライズアジャイルを進めているところである．

アジャイル開発の考え方とその際の出荷判定について事前に関係者に説明

3.　要件の検討と変更への適応

3.1　アジャイル開発の採用

本節ではアジャイル開発を採用した理由について記す．図4は，横軸は要求リスクを，縦軸は技術リスク，あるいは実装リスクを示す．また，各開発プロセスの模式図において，横軸は要件範囲を，縦軸はプロセスを示し，繰り返し開発する部分を緑色線の四角で表す．時間軸は示していない．

ウォーターフォール開発では，はじめに要件を定め，定めたすべての要件範囲に対して開発プロセスを進めていく．プロジェクト進行の時間軸はプロセスの縦軸と一致することになる．対して，アジャイル開発では要件範囲を非常に小さく分割し，優先順位の高い要件から開発を進めていく．プロジェクト進行の時間軸は，基本的には，要件範囲の横軸と一致することになる．ウォーターフォール開発は計画駆動型の開発であるため，要求リスクと技術リスクが低い場合に有効な方法であるのに対し，アジャイル開発は実測駆動型の開発であるため，要求リスクと技術リスクが高い場合に有効な方法であるといえる．

また，最近はハイブリッドアジャイルという言葉が登場している．これは，要件範囲全体を俯瞰してプロセスごとに開発するウォーターフォール開発の要素と，要件範囲を限定して繰り返し開発するアジャイル開発の要素とを組み合わせたプロセスである．ハイブリッドアジャイルには，プロジェクト特性とその目的に応じてさまざまなパターンがある．たとえば，要求リスクを低減するために，要件定義から設計の一部までを繰り返して実施し，その後は従来通りにプロセスを進めるパターンがある（図4右上，要求リスク対応型）．また，技術リスクや実装リスクを低減するために，要件定義から設計の一部までは従来通りにプロセスを進め，その後，残りの設

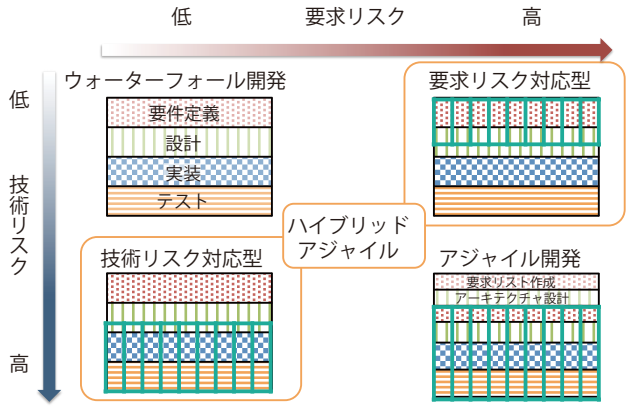


図 4　開発プロセスの選択

計から実装，テストまでを繰り返し実施するパターンがある（図4左下，技術リスク対応型）．場合によっては，これらのパターンを組み合わせることもできる．

このように，プロジェクト特性に応じて，現在はさまざまな開発プロセスが考えられるが，本プロジェクトでは，以下の理由により，アジャイル開発を採用した．

① 今回のプロジェクトは新規事業創出のプロジェクトである．リーン・スタートアップにおいて，価値仮説や成長仮説に基づいて顧客や市場に実際に製品を提供し，検証結果に応じて方向転換（ピボット）することが必要となる．このため，短いサイクルで動作確認でき，機能の変更や追加をしやすいアジャイル開発が向いている．

② 筆者らがこれまで実施してきたアジャイル開発事例は，プロトタイプあるいはウォーターフォール開発によるベースに機能を追加する事例が多かった．まったく仕様が見えていない新規開発の最初からアジャイル開発を適用する経験と実績を積みたいと考えた．

リーン・スタートアップには段階的に提供できるアジャイル開発を

3.2 MVP の検討とバックログの作成

リーン・スタートアップでは，最初に要となる仮説に基づいて実用最小限の製品（MVP：Minimum Viable Product，以下MVPと記す）を作る．MVPを実際に顧客に使ってもらい，提供する製品やサービスが本当に価値のあるものかどうかを検証する（価値仮説の検証）．次に，持続的に拡大していけるかどうかを検証する（成長仮説の検証）．検証した結果が期待した仮説と異なる場合は，そのまま進めるか，あるいは方向転換（ピボット）するかを検討する[7]．

MVPを検討するにあたっては，ユーザーストーリーマッピングを参考にした[8]．まず，水産養殖事業者や研究機関などの業界関係者へのヒアリング結果から，実現したい要件をユーザーストーリー（以下ストーリーと記す）として付箋に書き出した．次に，このストーリーを図5に示すように，ユーザが操作する時系列順（横軸）と優先度（縦軸）の2軸で整理した．図5において，同じ行に示したストーリーは，想定される一連の操作であることを示す．一般的に製品機能の検討では，多くの機能を盛り込みたくなるが，今回のMVPの検討ではできる限り開発する機能を削ぎ落とすことに注力した．図5の一番下の行に示されているストーリーがMVPとして削ぎ落とした機能である．プロダクトバックログによる要件の管理は，優先順位がシンプルに分かる一方で，ユーザが操作する時系列は分かりにくい．ユーザの操作順序を踏まえて優先度を考えることにより，要となる機能と付随的な機能を整理することができた．

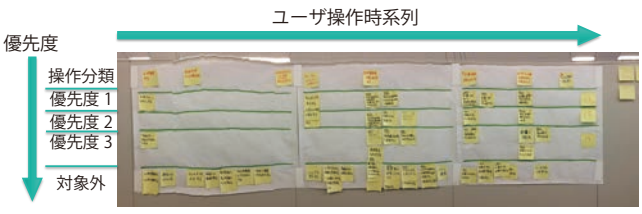
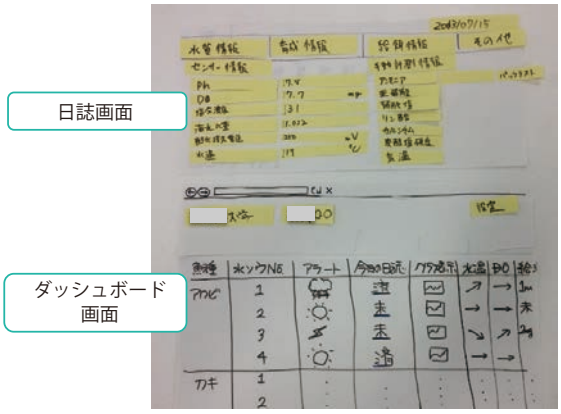


図5　ストーリーのマッピング

MVPは，ユーザ操作を考慮の上，機能を削ぎ落とすことに注力

MVPの機能を検討した後，これらの要件から，開発する製品の画面イメージをペーパープロトタイピングを用いて検討した（図6（a））．写真の上部が日誌画面，下部がダッシュボード画面の原型である．次に，作成したストーリーのマッピングとペーパープロトタイピングを持って，これまでヒアリングを重ねた水産養殖事業関係者を訪問した．そして，この時点でのフィードバックを



（a）ペーパープロトタイピング

（b）PowerPoint によるプロトタイピング

図6　プロトタイピング

いただき，それを踏まえてPowerPointによるプロトタイピングを作成した(図6（b）)．開発に際しては，プロダクトバックログを用いてストーリーを優先順位のみで管理し，各ストーリーについてはプランニングポーカーを用いて相対的にストーリーポイントを見積もった[9].

3.3　継続的評価によるフィードバック

開発している養殖管理ポータルを自らも試用し，評価しながら開発を進めるために，社内に水槽やポンプをはじめとする飼育環境を用意し，アワビの飼育を開始した．社内のアワビ飼育の様子を図7に示す．実際に社内でアワビを飼育することにより，ポンプの故障や害虫などによりアワビが死滅することも体験した．ユーザ体験を積む中で，週末や連休中も出社して飼育環境を維持することがいかに大変であり，かつ，重要であるかを理解することができた．水産養殖において飼育環境をモニタリングし，異常発生を検知して，適切に対応することが必要であることを，自分たちの経験を通して痛感した．現在，養殖管理ポータルではセンサにより，気温，水温，溶存酸素濃度，塩分, pHなどを計測しているが，今後は画像解析技術などを用いて養殖対象の動きをモニタリングすることで，より高度な水産養殖の管理システムを築きたい.

社内でアワビを飼育してユーザ体験し，自分たちでも常にシステムを評価

また，サービス提供の準備の段階からは，サービスデスクのメンバが顧客からの質問に適切に回答できるようにするために，アワビ飼育を引き継ぎ，サービスデスクのメンバが日常的に養殖管理ポータルやセンサ等を利用するようにした.

さらに，定期的に水産養殖事業者や研究機関をはじめとする業界関係者を訪問し，継続的に開発中のシステムへの評価をいただいた．訪問を継続する中で，「魚を正確に数えられるようにして欲しい」という新たな要求を得ることができた．この要求については，本製品とは別に，画像解析技術を用いて魚の数をカウントするNECフィッシュカウンターを開発した[10].

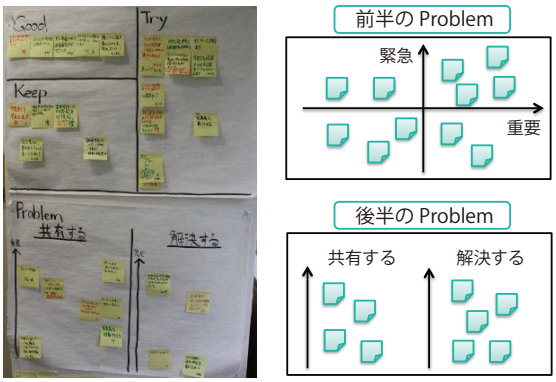
図7　社内のアワビ飼育の様子

養殖業者への継続的なヒアリングからNEC フィッシュカウンターを開発

3.4　継続的改善による適応

アジャイル開発は，プロジェクト早期から一連の開発プロセスを実施し，実際に起こる問題に対して改善を繰り返すことによって，プロジェクトを成功に導く方法である．2週間のスプリント終了時には，必ず，振り返りを実施し, プロジェクトの改善を図った．振り返りでは，KPTという方法を採用し，メンバ全員が①Keep：今後も継続する良いこと，②Problem：困っていることなどの問題，③Try：改善に向けたアクションを付箋に書き出し，共有した（図8）[11]．メンバ全員が付箋に書き出してから共有することで，全員が平等に発言することができた．今後も継続する良いことについては，筆者らは「Good：良いこと」と「Keep：今後も継続すること」に分けて運用した (図8 (b) 左写真)．振り返りの最初に，スプリントを終えて良かったことをGoodとして挙げることは，振り返りの場のアイスブレイクにもなった.

（a）振り返りの様子



（b）KPT ボード（左）とその変化（右）

図8　継続的な改善のための振り返り

プロジェクトを立ち上げた当初は，実際に非常に多くの問題が発生した．限られた時間で重要な問題に対象を絞って議論するために，KPTボードのProblemの領域は緊急度と重要度の2軸を用いて4つの領域に分け，緊急かつ重要な問題から対処していった．また，多くの問題に対処していき，プロジェクトが落ち着くにつれて，本プロジェクトとは直接的には関係ないが，個人的には困っているという内容もProblemとして挙がるようになった．このため，プロジェクトの後半はKPTボードのProblemの領域を「共有する」と「解決する」の2つの領域に分け，それぞれ重要なものから上に貼り出すようにした（図8（b）右図）．

さらに，プロジェクト当初の，開発が主な時期はプロダクトオーナー，開発・運用保守チーム，スクラムマスターで振り返りを実施していたが，運用保守やサービス提供が本格化してからは，サービスデスクのメンバも一緒に振り返りに参加するように改善した．サービスデスクは顧客からの問合せ等に対応するとともに，実際に社内でアワビを飼育しながら開発システムを継続的に利用し，評価している．このサービスデスクが振り返りに参加することにより，開発・運用保守チームがサービスデスクの問題や要望を知ることができる．サービスデスクのメンバが振り返りに参加することは，より使いやすいシステムを検討するために役立っている．また，サービスデスクとしても，要望を聞いてもらえるだけでなく，振り返りに参加して開発・運用保守チームとコミュニケーションを取ることにより，今後の機能追加の検討状況を知ることができるため，サービスデスクへの問合せに対する回答の対応にも活かせている．

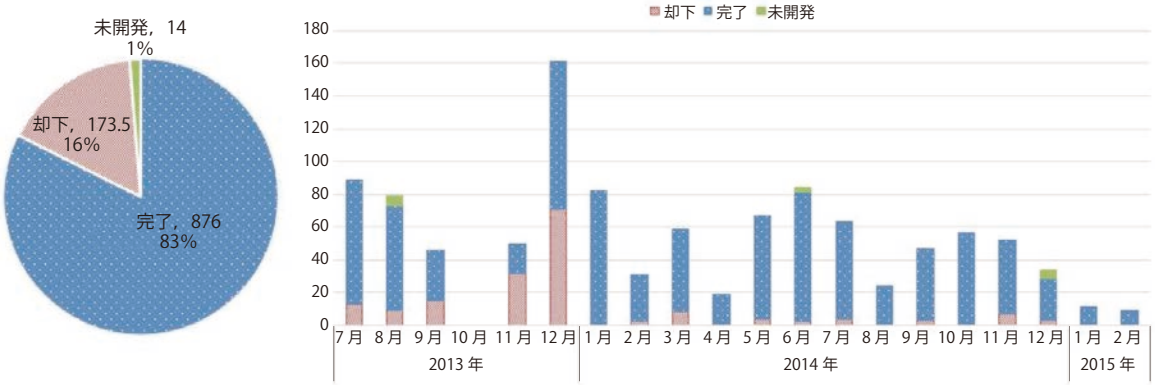
アワビを飼育してシステム評価する
サービスデスクも一緒に振り返り

3.5　アジャイル開発の適用評価

2015年3月時点においての要件の開発状況を図9（a）に示す．最終的に，プロジェクト期間中にプロダクトバックログに挙げた全要件の83%にあたる876SP(Story Point:見積り値)の開発を完了した．開発においては，優先順位の高い要件から開発をしていき，要求や要件が発生した際には，随時プロダクトバックログに追加し，優先順位の見直しを実施した．プロジェクトの進行に伴って状況が変わり，開発の必要がなくなった要件は「却下」として管理した．これは全体の16%である173.5SPに相当した．また，優先順位が低く，最終的に開発に至らなかった要件は1%の14SPであった．次にこれらの要求／要件が発生した時期とその最終的な開発状況を図9（b）に示す．図より，プロジェクトの初期に挙げた要件の中には最終的に「却下」とした要件が多く，プロジェクトの初期の要件は不安定であったことが分かる．さらに，要求および要件はプロジェクトの途中でも随時発生しているが，それらの要件についてもおおむね開発を完了し，変化に適応しながら開発を進めたことが分かる．顧客に実際に使用していただき，そこからフィードバックを得ながら，プロジェクト関係者がアイディアを出すことにより，顧客の目的をより的確に捉え，利用しやすいシステムへと改善しながら開発を進めることができた．

アジャイル開発を適用することで
随時発生する要求に適応できた

2.1節に記載したように，本システムは水産養殖において水質の状況と変化を定量的に把握し，日々の養殖の状況を記録してノウハウを蓄積し，養殖方法の改善を図るためのシステムである．現在の水産養殖において，このような取り組みは先端的であり，まだまだ一般的な方法ではない．それにもかかわらず，サービス開始後1年



（a）要件の開発状況　　（b）要求／要件の発生時期

図9　要件の変更と適応状況

の2015年12月時点ですでに10セット（クラウドサービスのほか，コントローラPC，水質センサを含む）を実際に稼働させていただくことができた．さらに，本システムを利用したことにより，①養殖の環境状況と変化をリアルタイムに監視することで死滅を未然に防ぐことができた事例，②降水予報の際に，かけ流し式から，ろ過循環式へと切り替えて，水温低下と濁りを抑えることができた事例など，本システムの利用が養殖方法の改善につながった事例も出始めている．これらの事例の詳細については参考文献を参照いただきたい[2],[12].

本システムを利用することで養殖方法の改善につながった事例が出始めている

第1章で述べた通り，アジャイル開発はリソースと期間を定めた上で，開発する要件の範囲を調整する方法である．よって，アジャイル開発ではQCD（Quality, Cost, Delivery）は基本的に固定であるが，このQCDについて本プロジェクトのプロダクトオーナーの所感を以下に記す．

Quality：

- 常に動くソフトウェアを確認でき，また，テスト自動化により常にデグレートがないことを確認でき，安心してプロジェクトを進めることができた．
- 運用を開始してからも重大なバグは実際に検出されていない．

Cost：

コスト，すなわち，生産性の指標としては，本来は単位時間当たりに開発した価値を用いるべきである．しかし，これは困難であるため，ここではデータを入手しやすい，人月工数当たりのライン数を用いた．本プロジェクトの人月工数当たりのライン数はウォーターフォール開発の当社標準と同等であった．なお，本プロジェクトでは，機能の追加と変更の繰り返しに伴い，適時リファクタリングを実施したが，比較においては最終的なソースコードのライン数のみを対象とした．

Delivery：

- 早期に開発に着手することができた．
- スプリントごとに生産性を計測し，それに基づいて計画を見直すことにより，優先順位の高い機能を期間内に実現できた．
- ほかの要因により，開発期間を急遽１カ月延長したが，プロダクトバックログから対応可能なストーリーをピックアップするだけで延長期間を有効に活用できた．

その他：

- 定期的なスプリントレビューとデイリースクラム（朝会）により，開発の進捗と課題や開発チームの状態が分かり，リスクコントロールがしやすかった．

プロダクトオーナーも QCD に満足

4. おわりに

開発プロセスの選択として示した図4は，いずれの開発プロセスで開発した場合も要件範囲は同じであるということを前提としている．しかし，アジャイル開発では顧客や市場から，あるいはプロジェクト関係者から常にフィードバックをもらいながら開発を進める．このため，図9（b）に示したように，プロジェクトの前半のみならず，プロジェクトの後半も含めて，常に要求と要件が変化することになる．よって，プロジェクト前半に要件を基本的に固定するウォーターフォール開発と，プロジェクト関係者から常にフィードバックをもらいながら開発を進めるアジャイル開発とでは，実際には要件の内容と範囲が厳密に同じになることはないと考えられる．

このように，アジャイル開発ではプロジェクトに参加したメンバや評価に参加したメンバ，プロジェクトに与えられた環境によって，要件の内容と範囲が変わり，開発する製品が変わる．実際に動くソフトウェアを見て，あるいは実際にユーザが自分でソフトウェアを動かし，使用してみることにより，ITシステムに詳しくないユーザでも新たに発想を広げることができる．アジャイル開発では，最終的にどのような製品になるかは分からない．そして，そこにアジャイル開発の利点があり，可能性があり，面白さがある．

さまざまなプロジェクトを遂行するベンダとして，プロジェクト特性に応じて開発プロセスを選択することは重要である．これまで経験を重ねてきたウォーターフォール開発に加え，今後もアジャイル開発の経験と実績を積み重ね，ハイブリッドアジャイルも含めた開発プロセスの中から最適な開発プロセスを選択していきたい．

参加メンバの関わり方が製品を変える．だから，アジャイル開発は面白い

謝辞　NEC養殖管理ポータルについてコンセプト検討の段階からさまざまなアドバイスをいただいた東京海洋大学の竹内裕准教授をはじめ，何度もヒアリングをさせていただいた水産養殖事業関係者の皆様に厚く御礼申し上げます．

参考文献

1) 誉田直美：ソフトウェア品質会計における品質要求と評価，情報処理，Vol.55, No.1, pp.58-64（2014）．
2) 竹内　裕，矢澤良輔，中谷貴子，小島郷史：情報通信技術を活用した陸上養殖施設の総合管理，養殖ビジネス 2014 年 12 月号，pp.10-13（2014）．
3) 柳澤尋輝，米今　彩，中尾勇介，野田尚志：アジャイル開発におけるユーザビリティ向上技術適用，人間中心設計機構・機構誌，Vol.10, No.1, pp.1-6（2014）．
4) ケン・シュエイバー，マイク・ビードル（著），スクラム・エバンジェリスト・グループ（訳）：アジャイルソフトウェア開発スクラム，ピアソン・エデュケーション（2003）．
5) Redmine Backlogs, http://www.redminebacklogs.net/（2016 年 2 月 22 日現在）
6) ケント・ベック，シンシア・アンドレス（著），角　征典（翻訳）：エクストリームプログラミング，オーム社（2015）．
7) エリック・リース（著），井口耕二（訳）：リーン・スタートアップ　ムダのない起業プロセスでイノベーションを生みだす，日経ＢＰ社（2012）．
8) ジェフ・パットン（著），川口恭伸（監修），長尾高弘（翻訳）：ユーザーストーリーマッピング，オライリージャパン（2015）．
9) ジョナサン・ラスマセン（著），西村直人，角谷信太郎（監訳），近藤修平，角掛拓未（訳）：アジャイルサムライ　達人開発者への道，オーム社（2011）．
10) 日経ビジネスオンライン キーパーソンに聞く：養殖の時代，いけすのマグロ 1000 匹はどう数える？, http://business.nikkeibp.co.jp/atcl/interview/15/238739/101500070/（2016 年 2 月 22 日現在）
11) 天野　勝：これだけ！ KPT，すばる舎（2013）．
12) 竹内　裕：岩手県大船渡市におけるアユ養殖と地方再生ビジネスモデル，養殖ビジネス 2015 年 12 月号，pp.56-60（2015）．

西潟 憲策（非会員）ken-nishikata@wh.jp.nec.com
2004 年 NEC ソフト（株）（現 NEC ソリューションイノベータ（株））入社．（一社）バイオ産業情報化コンソーシアム出向，産業技術総合研究所にてバイオインフォマティクス研究従事を経て，現在，アジャイル開発支援に従事．2004 年奈良先端科学技術大学院大学修士課程修了．2010 年同大学院大学博士課程修了．博士（工学）．認定スクラムマスター．PMP．

佐々木 裕之（非会員）hir-sasaki@tg.jp.nec.com
2007 年 NEC ソフト（株）（現 NEC ソリューションイノベータ（株））入社．システムエンジニアとして営業支援システムの開発に従事．水産 ICT のシステム構築の立ち上げ当社から開発メンバとして参画．2007 年弘前大学卒．

安藤 寿之（非会員）tos-andou@rs.jp.nec.com
1994 年日本電気ソフトウェア（株）（現 NEC ソリューションイノベータ（株））入社．官公庁システム開発業務を経て，現在，アジャイル開発支援に従事．認定スクラムマスター．Agile Japan 実行委員．

野沢 善浩（非会員）yos-nozawa@rf.jp.nec.com
2003 年 NEC ソフト（株）（現 NEC ソリューションイノベータ（株））入社．ユーザビリティ設計支援，アジャイル開発支援業務を経て，現在，情報セキュリティ推進に従事．2003 年東北大学大学院理学研究科博士課程修了．博士（理学）．PMP．

中谷 貴子（非会員）tak-nakatani@yk.jp.nec.com
1994 年日本電気ソフトウェア（株）（現 NEC ソリューションイノベータ（株））入社．システムエンジニアとして CRM（顧客関係管理）や ERP（統合基幹業務システム）などを担当，その中で同社が始めた新規事業創出プロジェクトにて水産養殖業を支援するビジネスモデルが選ばれ，水産 ICT を構築．水産養殖事業者，研究機関など業界関係者と共に「食の安心・安全・安定供給」を目的とした ICT 活用を展開中．1994 年横浜国立大学卒．

投稿受付：2015 年 11 月 5 日
採録決定：2016 年 5 月 26 日
編集担当：掛下哲郎（佐賀大学）