

PoC キャンバスを用いた機械学習システムの開発プロセスの管理手法

土肥 拓生

近年、機械学習に関する技術に注目を集めている。機械学習を利用したシステムにおいては、その企画から開発、運用まで、様々な種類の仮説検証が必要となる。一方、従来のソフトウェア工学の技術は、原則的に計画主導であり、不確実性を徐々に減らしていくものであるため、常に仮説検証が必要な場合には、そのまま適用することが難しい場合も多い。また、近年、主流になりつつあるアジャイル型の開発は、ビジネス上の仮説検証を考慮した開発手法と考えることができるが、機械学習を利用したシステムには、他の観点における仮説検証も考慮すべきであり、単純なアジャイル開発そのままでは解決できない課題も存在する。そして、このような機械学習を利用したシステムの場合には、仮説検証をその開発プロセスの中で管理する必要があると考える。著者は、その手法として、仮説指向ソフトウェア開発プロセス [3] を提唱している。本稿では、仮説指向ソフトウェア開発プロセスにおいて、仮説を整理、管理するための道具として利用する PoC キャンバスを提案し、その効果について議論する。

Recently machine learning are has been very focused. We have to do hypothesis testing for software systems with machine learning(machine learning system) in the whole process from planning to operation. Traditionally software engineering support us to make plans and reduce the uncertainty of the development. So it is difficult to apply the software engineering techniques to machine learning system. Although agile development which has been adopted all over the world can support to verify hypothesis on business, we need hypothesis testing of the other perspective. We have already proposed Hypothesis Oriented Software Development Process [3]. In this paper, we propose PoC Canvas which is useful for clarifying and managing the hypothesis.

1 はじめに

近年、機械学習に関する技術が注目を集めている。計算資源の高速化により深層学習 (ディープラーニング) などの技術が利用しやすくなり、実用的なシステムにおいても有益な結果が出せるようになってきたことが大きい。近年、機械学習に関する研究は非常に多く、機械学習システムに特有の課題があることも報告 [5] [4] されてきている。一方で、機械学習に関する研究は、新しい機械学習アルゴリズムの提案や、特定の領域において機械学習を適用したシステムの紹介が中心であり、ソフトウェア工学的見地から、いかに、機械学習システムを扱うべきかという議論は始まった

ばかりであるが、企業における事例に基づいたノウハウや課題として、機械学習システムを開発する手法、プロセスについて提示 [14] [13] されてきている。しかしながら、機械学習という確率的挙動を含むソフトウェアという一つの新しいパラダイムに対して、十分に議論されているとは言えない。

さて、一般的に仮説検証というプロセスが重要であるということは、以前より主張 [11] されてきている。近年主流となりつつあるアジャイル開発においても、ソフトウェア開発に関わる様々な不確実性を扱うために、仮説検証の考え方がその柱の一つとなっている。著者は、この仮説検証というプロセスを、機械学習システム特有の不確実性を扱うための開発手法として、仮説指向ソフトウェア開発プロセス (Hypothesis Oriented software development Process: HOP) を提唱している。本稿では、HOP における仮説検証を扱うための仕組みとして、PoC キャンバスを提案

する。PoC キャンパスは、機械学習システムの開発に携わるメンバーに、機械学習特有の仮説検証プロセスを容易に整理する手段を与えるとともに、何を仮説として捉えてソフトウェア開発を実施しているのか、仮説が否定された場合にどのように仮説の再構築を行うことが可能なのかを示唆することができる。

以降、2章で、HOP が想定している機械学習システムにおける特徴と、そのようなシステムの開発に必要な開発プロセスについて考察する。次に、3章で、機械学習システム、および、その開発の特性に基づき、独自の課題を解消するための開発プロセスであるHOP について示す。その後、3章で、HOP における仮説を監理するための PoC キャンパスについて提案するとともに、どのように PoC キャンパスが機能するかを示し、5章でまとめる。

2 機械学習システムとは

本章では、仮説指向ソフトウェア開発プロセスが対象とする機械学習システムの特徴について整理する。

2.1 機械学習システムの特徴

機械学習システムが持つ従来のソフトウェアシステムにはない特性は、これまで存在していなかった「仮説検証」が求められるということである。

ソフトウェア工学の歴史は、より多くの仮説検証を工学的に可能としてきた歴史である。初期のソフトウェア工学は、いかにユーザの要求を確定させ、その要求をいかに間違いなく実装するかというものであった。ユーザのニーズの多様化が進むとともに、環境の変化が加速するにつれて、全てを計画的に実施するソフトウェア開発がなかなかうまく行かなくなり、要求に対する仮説検証を実施する顧客開発 [2] という考え方や、それを実現するためにソフトウェア開発のプロセスにおける仮説検証を可能にするアジャイル開発 [1] が利用されるようになってきている。そして、より柔軟なシステムが求められるようになり、機械学習システムに注目が集まっているが、まだ未解決な課題も多い [7]。機械学習のモデルは、学習したデータに基づき確率的に決定されるため、そもそもシステムが期待したように振る舞うかを完全に論理的に保証

することは難しく、どの要素を特徴量とした扱うか、どの学習アルゴリズムを選択するか、どのように学習パラメタを設定するかについては、仮説検証を繰り返す必要がある。

さて、そもそもなぜ仮説検証が必要となるのであろうか。著者は仮説検証は、不確実性を扱うための有用な道具であることがその理由であると考える。この不確実性をいかに扱うかが機械学習システムの開発・運用には非常に重要である。本章では、著者が経験した、あるいは、見聞きした実際の機械学習を利用したプロジェクトの事例より抽出した機械学習システムの不確実性の特徴を示す。

2.1.1 プロジェクトの進捗とその不確実性

機械学習は、学習に利用するデータ、データのどの情報を特徴量として扱うか、どの学習アルゴリズムを選択するか、学習パラメタの設定によって、その結果として生成されるモデルの性質は異なる。そのため、従来のシステム開発のように、いかに詳細に設計したとしても、必ずしも期待したモデルを生成できるとは限らない。そのため、一般的には、利用可能なデータから期待するモデルを生成できるかという仮説検証から実施することが多い。そして、この課程のことを、Proof of Concept (PoC) と呼ぶ。

また、PoC ですら事前に予測不可能なことが多く、かつ、PoC にも膨大な時間やリソースが必要となる場合もあり、そのリスクを低減するために、PoC の事前検証のために、Pre-PoC という段階を踏むことも多い。Pre-Poc において、例えば、ごく一部のデータで検証させたり、代表的な手法での精度を計測することにより、より大規模なデータでコストをかけて PoC に取り組んだり、よりアルゴリズムのチューニングにリソースをかけるなどの妥当性を検証するのである。

ソフトウェアシステムとは、最終的にはユーザに対して価値を提供するものである。そのため、PoC により何らかの判断をある程度の精度予測するだけでは不十分である。また、その提供する価値がユーザにとって本質的なものでなければ、そのソフトウェアシステムが利用されることはない。また、モデルの振る舞いは確率的であるが、そのシステムで利用されるモ

デルが確率的な挙動を示した場合に、ユーザにどのような価値を提供できるかが重要である。例えばであるが、迷惑メールを判別するモデルを使ったシステムであれば、90%の精度で予測できれば、ユーザは満足するかもしれないが、自動運転の判断をするモデルの場合は、同じ精度で予測するシステムが搭載された自動車には誰も乗らないだろう。ここでは非常に分かりやすい例を提示したが、一般的なシステムにおいて、どのようなモデルをどのように利用すればユーザに価値を提供できるかは、非常に重要な仮説検証の対象である。

そして、全ての機械学習システムにおいて必要な訳ではないが、運用中に妥当な学習をし続けられるかというのも重要な仮説検証である。例えば、迷惑メールの例はこれが一歩いきやすいケースである。実際にユーザがメールを読むという操作(運用)の中で、扱うメールを学習することにより、よりユーザにとっての迷惑メールかどうかの判断に近づける可能性がある。しかし、一方で、実際の運用においては、どのような環境でどのようなデータを学習するか分からない状況において、適切な学習をし続けるための仮説検証をいかに扱うかは非常に大事なのである。

以上、まとめると機械学習システムのプロジェクトには、3つの不確実性を扱う必要がある。

1. PoCの実現性に関する不確実性
2. ソフトウェアにおけるモデルの不確実性
3. 運用における学習の不確実性

2.1.2 ソフトウェアの構造とその不確実性

本論文では、簡単のために、機械学習システムの構造を図1に簡略化して議論をすすめる。

この図において、モデルが学習モデルであり、確率的な挙動を示すものである。モジュールと記載した部分が、通常のソフトウェアモジュールを表現し、確率的な挙動をせず、プログラムの通りに動作するものである。これらが連携して動作するのがソフトウェアシステムであり、ユーザに価値を提供するとともに、その結果に対して、ユーザからフィードバックを得ることができる。また、そのフィードバックや環境から

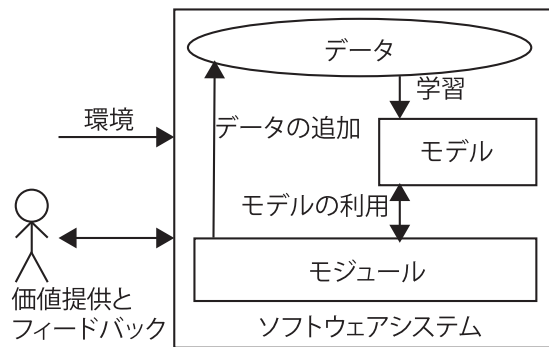


図1 機械学習システムの構造

の情報がデータに追加され、その変化したデータに基づいて、学習モデルが再構築される。

非常に簡略化した構造ではあるが、例えば、複数の学習モデルが組み込まれたシステムであったり、機械学習を利用しない部分が非常に複雑なシステムであったとしても、その機能ではなく、確率的な挙動を持つ部分とそうでない部分で分けることは可能である。そのため、議論に十分な汎用性を持っているはずである。

この図を、確率的な挙動をするモデルを中心に見ていくと、モデルの不確実性に誘発されることにより、次の課題が導かれる。

1. モジュールがモデルの不確実性を扱うことが可能
2. ソフトウェアがモデルの不確実性を吸収することが可能
3. ユーザや環境からの情報をデータとして追加による不確実性を扱うことが可能

最初の2つの課題は似ているが観点が異なる。例えば、雨の降りやすさを示すような値を提供するモデルがあったとしよう。学習モデルが分類モデルであれば想定外の結果を返すことはないが、回帰モデルであれば、その値域には不確定な部分が存在する。その場合に、モデルの結果を2倍したものが降水確率として利用していたとして、65という値をモデルが返してもモジュールとして整合性がとれるかというのが前者である。傘を持っていくべきかどうかを指示するソ

ソフトウェアだったとして、上記の 100%以上や、0 以下といった想定外の値だった場合に、何か問題が起きていると判断し「不安定なので注意してください」といった情報を提供したり、他の天気予報の情報を提供したりするというシステムとしての振る舞いが考慮されているかというのが後者である。もちろん、いずれの場合も何が例外的な事象なのかというのは規定されている必要がある。

2.2 機械学習システムのためのプロセス要件

では、ここまでで考察した機械学習システムを開発することに適した開発手法とは、どのようなものだろうか。別の言い方をすると、どのような開発手法であれば、プロジェクトに関する 3 つの不確実性を適切に扱い、ソフトウェアの構造に起因する 3 つの不確実性に対する課題に対処し、機械学習システムプロジェクトが成功させることができるのであろうか。

本章では、これらの観点から機械学習システムを開発する際に適した開発プロセスが満たすべき要件について検討する。

2.2.1 PoC の実現性に関する不確実性

PoC の実現性に対しては、一般的に不確実であることが認識されている場合が多い。しかし、プロジェクトとして考えた場合に、どれだけ多くの PoC を実施しても、どれだけ PoC で精度を上げることができたとしても、製品に貢献できなければ、ユーザに対して価値を提供できない。また、いくらリソース費やしたとしても、データによっては期待するモデルを構築できない可能性は十分にある。そのため、PoC としての成功、並びに、失敗を規定することにより、無駄にリソースを費してしまわないように注意を払う必要がある。

なお、機械学習を使わない手法との比較も重要である。

2.2.2 ソフトウェアにおけるモデルの不確実性

ソフトウェアにおいて、どのような不確実性が許容できるのかを把握することが必要である。また、PoC により未知の知識を得てそれを利用した画期的なシステムを構築したいという話を聞くことが度々あるが、そのようなアプローチを完全に否定することは

できないが、現時点での機械学習の効果から考えれば、要求工学や顧客開発など、ソフトウェア工学のこれまでの知見を十分に活用して、ユーザが求めているものを見定めた上で、それを実現するために欠けているものを機械学習で補えないかを検討する方が妥当であると考えられる。

すなわち、ユーザにどのような価値を届けるのかを見据え、その価値を提供するために欠けている情報を機械学習で補えるという仮説を立て、その仮説を PoC として検証する方が望ましい。

2.2.3 運用における学習の不確実性

運用における学習を実現するためには、そのソフトウェアの重要度に応じた対策が必要となる。前章で例示した迷惑メールの判別のような比較的重要度の小さいものであれば、数多くの学習データを収集できるような運用を検討する方が、最終的なユーザへ提供可能な価値は大きい可能性があるが、自動運転のような場合であれば、新しく学習したモデルが、少なくとも最低限の水準を満たしていることを保証する仕組みが必要であろう。この最低限の水準を保証するためには、製品自身に新しい学習モデルを評価する仕組みが内包させることなども考える。または、ユーザ自身に判断を仰ぐといった設計が適している場合もあるだろう。

いずれにせよ、新しく学習したモデルが、そのソフトウェアにとって最低限の水準を満たしていることを保証する仕組みが必要となる。

2.2.4 モジュールがモデルの不確実性を扱うことが可能

学習モデルがある入力に対して、どのような出力を返すかは、不確実であったとしても、その結果を利用する側のモジュールは期待した通りに動作することを保証する必要がある。このモデルの不確実性をモジュールが扱うためには、モジュールに対する自動テストと組み合わせることが最も容易であると考えられる。なぜなら、モデルの挙動は予測不能であっても、モデルの出力に対する期待する範囲は予想可能である。そのため、例外状態における自動テストを用意することで、モデルが想定外の結果を導き出した場合でも、モジュールとしては、想定された挙動を保証すること

が可能になる。

2.2.5 ソフトウェアがモデルの不確実性を吸収することが可能

前節で述べたように、モジュールが適切にモデルの不確実性を考慮されていれば、ソフトウェアとしては、モジュールの挙動は想定内に収まるため、その際の挙動について検討を行えばよい。ただし、前章の例で言えば、モデルが適切に学習できなかったために、常に「不安定なので注意してください」という表示をするようになってしまっていないかなど、継続的なソフトウェアとして妥当な動きをしているかを確認する仕組みがある方が望ましい。

2.2.6 ユーザや環境からの情報をデータとして追加による不確実性を扱うことが可能

この課題は、運用における学習の不確実性の扱いと共通する部分が多い。ソフトウェアの利用という観点からすると、利用において取得できたデータが全て有効なデータとは限らない。PoC や事前の学習であれば、エンジニアがデータ処理をすることが一般的であるが、運用後に取得したデータにおいては、恣意的に欠損値を埋めたりすることができないため、そのようなデータの加工についても十分に検証する必要がある。

3 仮説指向ソフトウェア開発プロセス (HOP)

本章では、2章で考察した特性を持つ具体的なプロセスを提案する。著者はこのプロセスを仮説指向プロセス (Hypothesis Oriented software development Process: HOP) と呼んでいる。HOP は、大きく PoC フェーズ、プロダクトフェーズ、運用フェーズの3つのフェーズから構成される。

3.1 PoC フェーズ

PoC フェーズは、その名前の通り PoC (Proof of Concept) を実施するフェーズである。HOP のにおいては、機械学習システムには、2種類の PoC が必要であり、2種類の PoC を反復的に実施するものである。

1つ目の PoC とは、機械学習においてよく言われる PoC である。あるデータがあり、そのデータを元に期待する機械学習モデルを構築することができる

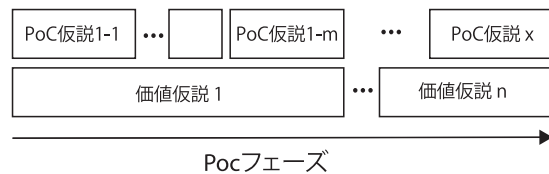


図2 PoC フェーズの概要

という仮説の検証である。この仮説を、HOP においては、モデル仮説と呼ぶこととする。

もう1つの PoC とは、ビジネス上の仮説であり、HOP では価値仮説と呼ぶこととする。価値仮説とは、あるモデル仮説が妥当と検証されれば、その機械学習モデルを用いて、ユーザに対してある価値を提供することが可能であり、かつ、ユーザもその価値を求めてくれるであろうという仮説である。このようにビジネス上の価値検証を並行して実施することが、HOP の特徴である。

一般的には、この2種類の PoC を次のような反復開発により実現する。

1. ユーザへ提供する価値仮説を策定
2. 価値仮説の前提となるモデル仮説を策定
3. 価値仮説を検証
4. モデル仮説を検証
5. モデル仮説の検証結果に基づき、以下のいずれかを選択
 - (a) 新しいモデル仮説の策定し、4に戻る
 - (b) 新しい価値仮説の策定し、2に戻る
 - (c) PoC フェーズ終了
 - (d) プロジェクト終了

図2に、PoC フェーズの流れのイメージを示す。この図で示すように、価値仮説とモデル仮説は同時に進むこととなる。

3.2 プロダクトフェーズ

プロダクトフェーズでは、モデル構築とプロダクト開発を実施する。このフェーズにおいては、アジャイルのフレームワークである Scrum [10] に基づいて開発を進める。本論文では、Scrum の詳細については省略するが、ユーザストーリーと呼ばれるユーザの体験ごとに機能を実装し、スプリントと呼ばれる期間を固

定した反復開発を実施するフレームワークである。

ただし、機械学習を使わないソフトウェア開発とは異なり、以下のことを考慮する。

- 直接的にユーザに価値を提供しないモデル構築のための要件もユーザストーリーとして扱う
- 運用フェーズを想定した学習モデルの妥当性の確認手法を確立する
- 機械学習システムの性質に応じた自動テストを記述する
- done の定義として、モデルが想定外の振る舞いをした場合のテストケースを記述することを規定する

特に確率的な挙動の機能に対する自動テストは手法を検討する必要がある。その手法は本論文の範囲ではないが、単体テストに代表される通常の自動テストの考えを拡張することも考えられる。例えば、一般的な単体テストであれば、ある入力に対して期待する出力が得られるかを検証するが、ある入力群に対して、期待する出力が得られた割合が一定以上あるといった検証などが必要となる場合がある。

3.3 運用フェーズ

運用フェーズは、プロダクトごとに検討する必要があるが画一的な手順を設定することは現実的ではないと考える。ただし、以下の点は考慮すべき点である。

- 運用中に取得できたデータにより学習をするかを確定する
- 学習する場合は、その妥当性をどのように検証するかを確定し、妥当でない場合にどうするかも規定する
- 運用中にソフトウェアのアップデートなどの介入方法について規定しておく

4 PoC キャンバス

HOP の PoC フェーズにおいては、価値仮説とモデル仮説の 2 種類の PoC を実施する。この仮説を同時に整理する手法が PoC キャンバスである。本章では、この PoC キャンバスを提案するとともに、HOP においてどのような効果があるのかを示す。

Project Name			
7: Value		8: Accountability	
5: Customer	1: Difficulty	2: Performance	3: Data
6: Problem	Alternative	Restriction	4: Algorithm
10: Cost		9: Maintainability	

図 3 PoC キャンバス

4.1 PoC キャンバスの定義

PoC キャンバスとは、図 3 に示す 10 の領域に区分された表を埋めながら、PoC を整理する道具である。PoC キャンバスの左側には、価値仮説に関する情報を記載し、右側にはモデル仮説に関する情報を記載する。

以下、PoC キャンバスの各領域について述べるが、各領域に付記した数字の順番に埋めていくことが容易であると思われる。まず、Difficulty の領域を検討することにより、対象とするシステムにおいて、機械学習が必要となる課題を明確にし、それに基づいて、Performance の領域を明確にすることにより、Difficulty を解消するために必要な機械学習モデルの要件を規定する。そして、PoC 仮説を中心に検討し、その Performance を実現するための Data と Algorithm に対して仮説を立てる。一方で、価値仮説を明確にするために、Customer とその Problem を明確にし、Value を記述することで、仮説上の矛盾の有無を確認する。その上で、Poc 仮説と価値仮説をつなぐための Accountability や、Maintainability、Cost を整理することで、プロジェクトとして成立しうるかを検証可能とする。

ただし、場合によっては、機械学習の実運用の企画のために、PoC 仮説から入る場合もあるだろう。その場合には、利用可能なデータについて検討して、Data の項目から記述し、Algorithm、Performance と規定しなければならない場合もあるだろう。そして、その機械学習モデルがあれば、解決できる Difficulty を探

索し、それを元にプロダクトの企画を構築することも可能である。

大事なことは、最初の時点で全ての項目を記述できる必要は必ずしもないかもしれないが、この PoC キャンパスの全項目を記述し、その検証を行なっていない場合は、そのことを認識しながら、プロジェクトを進める必要があるということである。検証できていないということは、将来的にその仮設に対して、否定的な結論に至る可能性があるということであり、その場合は、キャンパス上の周囲の領域を中心として、変更を加えなければならないということである。

例えば、Performance の領域において、想定した精度の学習モデルの構築が難しいということが判明した場合には、他のデータや、前処理を工夫するなど、Data の領域を変更する、または、より精度が期待できるアルゴリズムに変更するように Algorithm の領域を変更する、もしくは、Difficulty を調整して、対象とする価値仮設を変更することを検討しなければならない。

4.1.1 Difficulty

この領域には、HOP においては、価値仮説の中核をなす記述であり、顧客価値を実現するために機械学習により解決しようとする困難な点である。この項目を検討する際には、併せて、機械学習を利用しない場合には、どのような代替手段が存在しうるかを記述する。代替手段で十分だと判断された場合には、機械学習を利用しないシステムとすると判断する可能性も存在する。

4.1.2 Performance

この領域には、PoC 仮説として求められる機械学習モデルの性能を記述し、PoC 仮説の中核をなす。ここで言う性能とは、その機械学習モデルが見たすべき、予測精度や、計算時間などのことである。また、この機械学習モデルの対象外としてよい場合があれば、併せて記載しておく。PoC 仮説において、この記載が実現できないと判断された場合には、Data、Algorithm の変更、あるいは、価値仮設の Difficulty の変更が検討される。

4.1.3 Data

この領域には、PoC 仮説としての機械学習モデルの構築、及び、評価に利用するデータについて、取得経路、データのクレンジング、特徴量選択などの手法についてもこの領域に記述する。

4.1.4 Algorithm

この領域には、PoC 仮説としての機械学習モデルの構築に利用する学習アルゴリズムについて記述する。

4.1.5 Customer

この領域には、価値仮説として、提供する価値を必要としているであろう顧客について記述する。この領域の記述の粒度については、対象などにも依存するが、価値仮設の検証という観点からは、ペルソナ [9] まで抽出できていると望ましい。

4.1.6 Problem

この領域には、価値仮説として、顧客が抱えているであろう課題を記述する。この領域は具体的であればあるほど、Difficulty の妥当性、及び、価値仮設の検証が現実的かつ、迅速に実現することができる。

4.1.7 Value

この領域には、価値仮説の概要を記述する。

4.1.8 Accountability

この領域には、PoC 仮説として、機械学習モデルに求められる説明可能性について記述する。ドメインによっては、そのドメインにおける説明が可能であることが重要な場合がある。そのような場合には、その点を明記しておくことが重要である。

4.1.9 Maintainability

この領域には、PoC 仮説として、機械学習モデルに求められる保守性について記述する。

4.1.10 Cost

この領域には、価値仮説として、プロジェクトに対する期間や費用に対する制約を記述する。

4.2 PoC キャンパスの例

4.3 キャンパスについて

これまでもキャンパスという概念は提案されている。代表的なものとしては、ビジネスモデルキャンパス [8]、リーンキャンパス [6]、仮説キャンパス [12] などがある。PoC キャンパスも、これらのキャンパ

スの考え方に影響を受けているが、これらのキャンパスは、検討するビジネスとそれを取りまく環境を整理することを目的としてキャンパスである。PoC キャンパスは、機械学習システムには、確率的な挙動をする機械学習モデルと、ビジネス上の目的とを同時に監理するものである。

5 まとめと今後の課題

本稿では、HOP で利用する価値仮設と PoC 仮設を同時に管理するための PoC キャンパスを提案した。機械学習システムの開発は、産業界においては、徐々にノウハウは蓄積されてきているものの、適切に開発を管理するための体系的な手法は確立されているとは言えない。その理由の 1 つは、ソフトウェア工学においてこれまで扱ってこなかった種類の不確実性を扱うためである。PoC キャンパスにより、仮設を記述し、可視化することで、より適切な管理の手助けになると信じる。

現時点では、PoC キャンパス、及び、それを利用した HOP については、概念的に検証したに過ぎない。今後は、過去の機械学習プロジェクトをケーススタディとして、HOP、及び、PoC キャンパスの有効性をより具体的に示すとともに、その内容をより洗練させていく予定である。

参考文献

- [1] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D.: Manifesto for Agile Software Development, 2001.
- [2] Blank, S., Blank, S., and Dorf, B.: *The Startup Owner's Manual: The Step-by-step Guide for Building a Great Company*, K&S Ranch, Incorporated, 2012.
- [3] Doi, T.: 機械学習を利用したシステムにおける仮説検証を考慮した開発プロセスに関する考察, 日本ソフトウェア科学会 第 35 回大会講演論文集, 2018.
- [4] Kim, M., Zimmermann, T., DeLine, R., and Begel, A.: The Emerging Role of Data Scientists on Software Development Teams, *Proceedings of the 38th International Conference on Software Engineering, ICSE '16*, New York, NY, USA, ACM, 2016, pp. 96–107.
- [5] Masuda, S., Ono, K., Yasue, T., and Hosokawa, N.: A Survey of Software Quality for Machine Learning Applications, *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, April 2018, pp. 279–284.
- [6] Maurya, A.: *Running Lean: Iterate from Plan A to a Plan That Works*, O'Reilly Media, Inc., 2nd edition, 2012.
- [7] Mitchell, T.: The discipline of machine learning, Technical Report CMU ML-06 108, 2006.
- [8] Osterwalder, A. and Pigneur, Y.: Business Model Generation, Feb 2013.
- [9] Pruitt, J. and Grudin, J.: Personas: Practice and Theory, *Proceedings of the 2003 Conference on Designing for User Experiences, DUX '03*, New York, NY, USA, ACM, 2003, pp. 1–15.
- [10] Schwaber, K. and Beedle, M.: *Agile Software Development with Scrum*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.
- [11] 内田和成: 仮説思考 BCG 流 問題発見・解決の発想法, 東洋経済新報社, 1st edition, 2006.
- [12] 新井 剛市谷 聡啓: 翔泳社, 1st edition, 2018.
- [13] 本橋洋介: 人工知能システムのプロジェクトがわかる本企画・開発から運用・保守まで, 翔泳社, 1st edition, 2018.
- [14] 太田満久, 井上佳, 今津義充, 中山英樹, 上総虎智, 山裕市, 蘭頭隆太, 草野隆史: 失敗しないデータ分析・AI のビジネス導入: プロジェクト進行から組織づくりまで, 森北出版, 1st edition, 2018.