

アジャイル開発において最小効率で仕様変更を行う事を目指して

三菌 義隆[†] 大場 充[‡]
 広島市立大学[†] 広島市立大学[‡]

1. 背景

本研究ではソフトウェアのライフサイクル中の生産性を向上させるためにアジャイル開発手法を用いる。アジャイル開発手法は従来型の開発手法の問題点を解決することができると提唱されている。しかしながらアジャイル開発手法自体にも定番として受け入れられた定義がなく、故に属人性が高いなどの問題点がある。これらの問題点の為に、アジャイル開発手法の有用性は明らかなものの、一定規模以上のプロジェクトへ適用するのは難しいとされている。

米 Microsoft が行った大規模な調査 [1]によると、実際のプロジェクトへのアジャイル開発手法の採用は知名度と比較して少なく、また大規模開発へのアジャイル開発手法適用は困難との見解が示されている。一部の専門家はこれらの課題に対して形式化 (formalism) が有用と主張している。その内の 1 人である S.Wolff 氏は論文の中で形式手法 (formal methods) とアジャイル開発手法の 1 つであるスクラムと組み合わせるアイデア [2]を紹介している。Wolff 氏によれば、少なくとも理論上は、形式手法によりアジャイル形式手法の問題点を補うことができ、またソフトウェアの完成度を上げること一つまりは開発工程の繰り返しの削減ができるという事を確認したとの事である。

我々は判断木という新しい手法を提案する。この提案では仕様変更を形式的に扱い、かつソフトウェアをアジャイル的に開発、変更することにより、ソフトウェア・ライフサイクルに関わるコストを減少させることを目指す。

英題: Aiming to determine the change of specifications in the most efficient way in Agile Development

[†]: Yoshitaka Misono (Hiroshima City University)

[‡]: Ohba Mitsuru (Hiroshima City University)

2. 提案手法

本研究では判断木という特殊な木を用いる。この判断木にはソフトウェアのどの部分を変更するかという疑問文が述語という形で各頂点に書かれている。利用者は判断木に「社会システム」の形式的仕様、「社会システムの変更部分」の形式的仕様、「開発ソフトウェア」の形式的仕様を入力すると、出力として「ソフトウェアの仕様のどの部分が変更されるか」という情報を得ることができる。

3. 定義

本研究で用いる定義を述べる。読者は大学基礎レベルの集合論の知識を持っていることが望ましい。グラフ理論の知識も用いるが、随時説明を加える。

本来であれば本研究で用いる全ての用語について定義を記述すべきであるが、紙面の都合上グラフ理論で一般的な定義については一部割愛する。

有向木：有向グラフ D において、 $V(D)$ に空でない元の集合 $\{v_p\}$ が存在し、 $\{v_p\}$ の元でない $V(D)$ の任意の元 v_i に対して v_i と連結する $\{v_p\}$ の 1 つ以上の元が存在し、 $\{v_p\}$ の元から v_i への道に閉路を含まない有向グラフ D を有向木と呼ぶ。

判断木で使用する定義

論理変数：ある述語の真偽値を確定するための条件を与える変数を論理変数という。

判断：ある述語に定義された論理変数を束縛する値の組に対して真偽値を返す当該の述語が定義された有向木 T の頂点を判断と呼ぶ。

判断木：判断の集合を $\{v_i\}$ とおく。 v_i には 1 つ以上の葉と 1 つ以上の判断が存在し、 $\{v_i\}$ の集合を構

成するグラフが根付き有向木である時、その有向木 T を判断木 T_d とよぶ。

判断木の葉：判断木 T_d における葉を判断木 T_d の葉とする。

判断木の根：判断木 T_d における根を判断木 T_d の根とする。

判断木の定義より、判断木 T_d に含まれるすべての判断は定義可能な有限個の述語及び、その組み合わせによって構成される。述語またはその組み合わせを要素として構成される判断の集合は有限集合である。

4. 定理

判断木の根底となる定理を記述する。2つの補助定理と3つの定理を記述する。補助定理は定理の証明に使われる。全ての判断木は3つの定理全てを満足する必要がある。

補助定理(1)：任意の判断木 T_d において、根 V_r から葉の集合 $\{L_t\}$ に属する任意の葉 V_l に至る道が存在する。

解説：根から任意の葉に至れることを保証する。

補助定理(2)：任意の判断木 T_d に含まれる任意の述語 $p_i(i = 1, 2, \dots, n - 1)$ の真偽値を決めるため、当該の述語 p_i を除く1つ以上の頂点に対応する述語の集合 $\{p_i\}$ を必要とするとき、その p_i の真偽値を確定するために必要な述語のみから構成される頂点の集合 $\{p_i\}$ を木構造 S と置く。このとき述語 p_i と木構造 S を判断木 T_d の1つの頂点に対応する論理式 p'_i として書き換えることができる。

解説：ある頂点から葉まで伸びている部分木はそれと等価な1点の頂点にまとめられる。

定理(1)：任意の判断木 T_d において、その根 V_r から有限回の手続きで、必ず T_d の葉の集合 $\{L_t\}$ に属する特定の葉 V_l に至ることができる。

解説：判断木で処理を開始したとき、必ず処理が

有限時間内に完了することを表す。

定理(2)：任意の判断木 T_d に関して、その真偽値を確定するために必要な条件の要素を指定した複数の述語が与えられているとき、それらの述語から構成される論理式によって確定する判断木 T_d 上の任意の道の評価値は各述語の評価順序に関係なく、同じ述語の真偽値に対して同じ道の評価値を与える。

解説：チャーチロッサー性への言及。

定理(3)：もし判断木 T_d がコンパクトならば、 T_d は妥当性を満たす。

解説：判断木を構成するための情報が過不足なく与えられていることをいっている。

5. 結論

適切な判断木が与えられたとき、既存のソフトウェアシステムに対して仕様要求があったときに、仕様書に変更が必要か否か判断することが可能であると証明できた。

6. 今後の課題

理論のさらなるブラシアップが必要。特に定理3の「妥当性」に関して理論的根拠を与える必要がある。

参考文献

- [1] B. Murph, C. Bird, T. Zimmermann, L. Williams, N. Nagappan and A. Begel, "Have Agile Techniques been the Silver Bullet for Software Development at Microsoft?," in *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2013.
- [2] S. Wolff, "Scrum Goes Formal: Agile Methods for Safety-Critical Systems," in *Formal Methods in Software Engineering: Rigorous and Agile Approaches (FormSERA)*, 2012.