

# ユーザメモソフト my\_help のビヘイビアテスト開発

関西学院大学 情報科学科 西谷研究室 2535 那須比呂貴

## 1 目的

プログラム開発では、統合開発環境がいくつも用意されているが、多くの現場では、terminal 上での開発が一般的である。ところが、プログラミング初心者は terminal 上での character user interface(CUI) を苦手としている。この不可欠な CUI スキルの習得を助けるソフトとして、ユーザメモソフト my\_help が ruby gems に置かれている。しかし、Ruby gems として提供されているこのソフトは、テストが用意されていない。今後ソフトを共同開発を進めていくには、仕様や動作の標準となるテスト記述が不可欠である。本研究の目的は、ユーザメモソフトである my\_help のテストを BDD を用いて開発することである。

## 2 方法

■BDD ビヘイビア駆動開発 (Behaviour-Driven Development : BDD) は、テスト駆動開発 (Test-Driven Development : TDD) の工程への理解を深め、それをうまく説明しようとして始まった。TDD の持つ単語のイメージが構造のテストを中心とすべしというのに対して、BDD はソフトの振る舞いに中心をおくという意図がある。この違いが、初めに考えるべきテストの性質を変化させ、構造ではなく振る舞いを中心にテストを構築するという意識をもたせてくれる。さらに、実世界における開発チームやテストチーム、あるいはドキュメントチーム間のコミュニケーションの取り方を、システムで提供するものが BDD のフレームワークであり、Cucumber と RSpec はこれを実現する一つのシステムとして提供されている [1]。

■Cucumber と RSpec 図 1 は RSpec と Cucumber の関係図である。BDD の流れとして、まず Cucumber で一つのシナリオに焦点を当てて、その振る舞いを記述する feature を書く。一つの feature が書けたら、次に、それぞれ feature を実現するステップに分けて仕様を決める。これは TDD の red green refactoring の前に行う作業、「仕様を決める」に対応している。このプロセスが終了したら、RSpec に進む。RSpec では実際にテストコードを書き、ここでも red, green, refactoring をおこない、RSpec が成功したら、Cucumber の refactoring をおこなう。

## 3 ビヘイビアテスト駆動開発の実践

Cucumber と RSpec を用いて BDD で my\_help のテスト開発を進めていった。ここでは、焦点を合わせた my\_help の中での一つの振る舞いである「todo の更新」を例として詳し

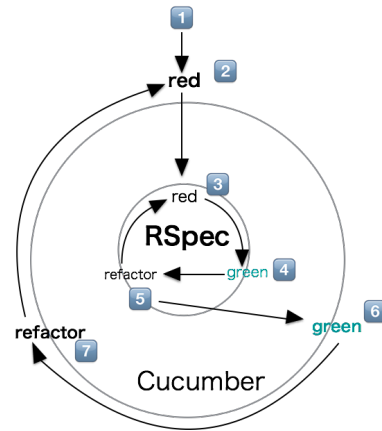


図1 RSpec と Cucumber の Red-Green-Refactoring サイクル間の関係、く見ていく。

- cucumber
1. まず、「todo の更新」のシナリオを feature で記述。
  2. cucumber を実行するとステップの元なるコードブロックが自動生成されるので、それをコピー&ペーストしてステップを記述。
  3. ここではあれば良いと思うコードを記述し、もう一度 cucumber を実行。

rspec

ここでは、エラーが出るはずで、そのシナリオに焦点をあて、RSpec に移行してテストコードを記述。

cucumber RSpec が成功したら、もう一度 Cucumber に戻り、red を green に。

このようにして、BDD のコンセプトに従って、my\_help の仕様や動作の標準を確定していった。

## 4 まとめ

my\_help のテスト記述の進展に伴って、仕様や動作の標準が確定した。my\_help は今後も開発者、ならびに多くのユーザの使用を通じて、どのように進化させれば便利なのかが徐々にわかってくる。つまり、ビヘイビアの記述により、今後 my\_help を進化させるための共同開発が円滑に進める手助けになる。さらに、my\_help の使用方法が明確になったことで、本研究で作成した my\_help の features を読めば初心者でも my\_help の振る舞いが容易に理解できるようになった。

## 参考文献

- [1] "The RSpec Book", David Chelimsky *et al.*, (翔泳社, 2012).