トップエスイー修了制作

コーディングルール適用による複雑度低減の試み

NECソフト株式会社　西潟　憲策　nishikata-kensaku@mxr.nes.nec.co.jp

開発における問題点

アジャイル開発は，仕様変更や機能追加を前提とするため，“シンプルな設計”にすることが特に強く求められる．しかし，実際のプロジェクトにおいて，仕様変更や機能追加を繰り返しながら“シンプルな設計”を維持することは非常に難しい．

手法・ツールの適用による解決

ソフトウェア設計を改善するためのコーディングルールをプロジェクトに適用することにより，ソフトウェアメトリクスの循環的複雑度（クラス平均複雑度およびメソッド平均複雑度）を低減した．

ソフトウェア設計を改善するためのコーディングルール

ソフトウェア設計を改善する9つのステップ＊

メソッド粒度の最小化

1. **インデント1段階**
   1つのメソッドにつきインデントは1段階までにする．
2. **else句禁止**
   else句を使用しない．

クラス粒度の最小化

6. **小エンティティ**
   すべてのエンティティを小さくする．
7. **インスタンス変数2つ**
   1つのクラスにつきインスタンス変数は2つまでにする．

オブジェクト責務の最適化

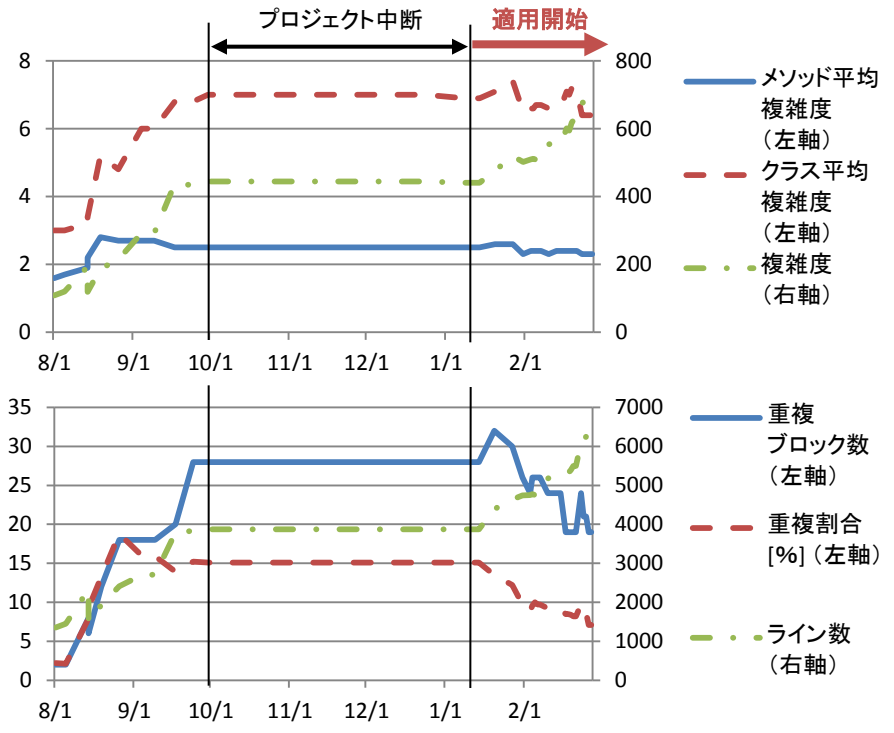
3. **プリミティブ禁止**
   すべてのプリミティブ型と文字列型をラップする．
4. **ドット1つ**
   1行につきドットは1つまでにする．
5. **名前省略禁止**
   名前を省略しない．
8. **ファーストクラスコレクション**
   ファーストクラスコレクションを使用する．
9. **プロパティ禁止**
   Getter/Setter プロパティを使用しない．

＊ ThoughtWorks Inc. (著) , ThoughtWorksアンソロジー：アジャイルとオブジェクト指向によるソフトウェアイノベーション　第5章「オブジェクト指向エクササイズ」 より

製品コード全体における随時適用

プロジェクトにおいて，3つのルールを随時適用*した．

＊ 仕様変更や機能追加において，コードを追記/修正する際に適用



パッケージコードにおける完全適用

3つのルールを完全適用し，加えて，「小エンティティ」を適用した．

	適用前	3つのルール適用後	小エンティティ適用後
メソッド平均複雑度	3.5	3.1	2.9
クラス平均複雑度	42	41（変化なし）	18.3

まとめと課題

- 複雑度が増加傾向であったのに対し，3つのルール適用により，複雑度をある程度低減することができた．
- クラス平均複雑度は，3つのルール適用だけでは十分に低減できず，「小エンティティ」適用によりクラス分解を促すことで低減できることを確認した．
- クラス粒度を最小化してクラス平均複雑度を低減するために，「小エンティティ」の適用をいかに低いコスト（障壁）で現場に導入するかを検討する必要がある．