

JaSST' 14Tokyoソフトウェア開発ライブ パネル 2014年3月7日（金）

富士ゼロックス株式会社 ソリューション・サービス開発本部
秋山 浩一

自己紹介（秋山浩一：@akiyams924）

- 1985年4月 富士ゼロックス入社
 - ◆ 現在はHAYST法のコンサルティング業務に従事
- NPO ソフトウェアテスト技術振興協会（ASTER） 理事
 - ◆ JaSST東京実行委員（2003年～2014年）
日本最大のテストシンポジウム1600名の動員
 - ◆ JSTQBステアリング委員（2006～）
テスト技術者資格認定を行う国際組織日本支部
- 日科技連 SQiP研究会 委員長（2011年～2013年）
- Wモデル研究会 主査（2011年7月～）
 - ◆ 電通大 西康晴先生、NEC 吉澤智美氏、MRT 鈴木三紀夫氏
- ISO/IEC/IEEE JTC 1/SC7 WG26委員（2009～）
ソフトウェアテストの国際標準ISO29119策定中
- 共著書『ソフトウェアテストHAYST法入門』（2008年度 日経品質管理文献賞受賞）、著書『ソフトウェアテスト技法ドリル』
- ソフトウェアシンポジウム最優秀論文賞受賞（2014年）
- 博士（工学）



①設計

（プログラムコード）

によって

テストは変わる

②これまでのTDDを
進めるだけで
エレガントな設計は
生まれるのか？

良い設計が
テストを
少なくする

テストが多いと
安心する？

設計に
応じた

テストを

用意すべき？

モヤモヤ①（設計とテストの関係）に対しての深掘り

設計を自由に

変更できるように、

評価は開発と独立

したほうが良い？

エレガント

な設計が

望ましい？

エレガントなコードは

トリッキーで、理解が困難

- 理解性が悪い

 - デバッグが困難

- HWが倍速していくのなら

機能性や効率性より、移植性や

理解容易性の方が大切

No.	どっちやねん？	
① -1	良い設計が テストを削減	テスト多いと安心
① -2	設計に応じたテスト	設計を自由に変更できるように、評価は開発と独立すべき
②	エレガントな設計が望ましい	エレガントなコードはトリッキーで、理解が困難 <ul style="list-style-type: none">• 理解性が悪い<ul style="list-style-type: none">→ デバックが困難• HWが倍速<ul style="list-style-type: none">→ 機能性や効率性より、移植性や理解容易性

No

TDD導入のメリット

- ① 開発のスピードアップ
→ テストは仕様の形式化？
- ② テスト容易化設計になりやすい
→ APIの設計が良くなる！
- ③ プログラムの変更耐性向上
→ テストがあるので凡ミス
を恐れずに内部設計変更できる

FUJI XEROX

