

7.1. ユニットテスト項目を少なくする必要性

Q. 何故、ユニットテスト項目を少なくする必要があるのか？

A.
ソフトウェア品質向上のため
→ **ソースコードの品質向上が必要**

何故ユニットテスト項目を少なくする必要があるのでしょうか。

その理由は、ソフトウェア品質向上のためです。
そのためには、ソースコードの品質向上が必要です。

ソースコードの品質向上は、ソフトウェア品質向上にどのような効果をもたらすでしょうか。

- ユニットテスト工程の工数が削減される
(ユニットテストは基本的に全ステップ対象であるため、時間が莫大にかかるが、項目が減れば時間も減ることになる)
- ステップ数あたりの欠陥数が削減される

ユニットテスト工程の工数削減

コーディングによって、ユニットテスト項目を減らすことが可能です。
ユニットテスト項目を少なくすると、工数削減につながります。

1. 工数の削減は、テストの精度を上げ、欠陥の検出漏れを減らすことになります
→ユニットテスト工程で検出すべき欠陥を、後工程に持ち越さない
2. ユニットテスト設計・実施時間を減らすことになります
→ユニットテスト工程を期間内に完了し、後工程の開始を遅らせない
→後工程を圧迫しない
→後工程で品質確保に必要な工数を割り当てることができる

ユニットテスト項目を少なくするコーディング
↓
ユニットテスト工程の工数削減を意識したコーディング

注：ユニットテスト項目設計をサボっても項目数は減りますが、それでは品質低下、欠陥の流出を招くことになります。

ステップ数あたりの欠陥数削減

ステップ数あたりの欠陥数削減とは

ユニットテスト項目数		最大欠陥件数	
100件	➡	100件	} 品質の差 2倍
50件	➡	50件	

上の表はユニットテスト項目数と品質の関係を表したもので、
"1000stepのソースコードに対し、100件のユニットテスト項目設計したなら、100件の欠陥があるかもしれない"
"1000stepのソースコードに対し、50件のユニットテスト項目設計したなら、50件の欠陥があるかもしれない"
ということを表しています。
単純計算しても品質が2倍違うということから、コーディングによって、欠陥が発生しやすいポイント
つまり、ユニットテスト項目になる箇所を減らすことが可能と考えられます。

ユニットテスト項目を少なくするコーディング
↓
欠陥が発生しやすいポイントが少ないコーディング

注：ユニットテスト項目設計をサボっても項目数は減りますが、それでは品質低下、欠陥の流出を招くことになります。