2024年9月23日　19:25

実装Input

開発は、品質に不安を感じているところについて、その理由や動作メカニズムを評価チームに伝える
評価は、コードレビューや実装内容の定例会など、従来は開発だけの場に参加し、内部動作の複雑さを知る
実装Inputにより、評価は開発から内部動作のメカニズムや複雑性を学び、開発は評価からのFB（例. 過去不具合事例)から設計に関わる考慮漏れに気づける。

触りこみ

開発は、実装したソフトをできるだけ早く（例. 毎日）リリースする。
評価は、リリースされたコードをすぐに（例. コミットされた翌日）に触りこむ。
触りこみにより、評価は動作メカニズムを具体的に理解し、多義性不具合を早期に発見することができる。

- **多義性不具合解析**

開発は、多義性不具合を優先度上げて解析して根本対策（例. 設計見直し）、発生メカニズムを評価に伝える。
評価は、多義性不具合の発生メカニズムを理解し、類似不具合がないか確認する。
多義性不具合解析により、不具合を発生させる動作メカニズムを理解し、開発は不具合の根本対策（例. 設計見直し）、評価は類似不具合確認を実施することで、物件の品質リスクを下げることができる。

バウンダリオブジェクト・意味化

バウンダリオブジェクト：多義性不具合
意味化：多義性不具合は品質リスクが高い未知の一部が表面化したものであり、その動作メカニズムを解析して理解することで品質リスクを下げられる、ということを共通見解にする

バウンダリオブジェクト：品質リスク一覧
意味化：品質リスクを見える化し、その解釈（品質リスクの高さ）、優先度、対応方針を共通見解にする。