

Übung: 'prospective memory' training

Ziel: Die Fähigkeit zum vorausschauenden Erinnern entwickeln - vorausschauend planen, sich an etwas zu erinnern (wenn ein bestimmtes Ereignis eintritt).

Vorgehen: Wähle morgens zwei bis drei Einträge aus der folgenden Liste:

BugReg	Wenn ich heute zum ersten Mal	eine Fehlermeldung mit einer Stacktrace bekomme	dann lese ich im Katalog der Standard-situationen nach und schreibe mindestens einen Test-case.
CodDup		duplizierten Code sehe	
CProp1		eine Methode mit verschachtelten Kontrollstrukturen sehe	
CProp2		eine lange Methode (größer als eine Bildschirmseite) sehe	
CProp2		eine große Klasse (mit mehr Methoden, als die Outline anzeigen kann) sehe	
DelgIn		Vererbungshierarchien sehe, die nur zum Code-Sharing gedacht sind	
DelgIn		sehe, daß eine Klasse das Verhalten der Oberklasse unterdrückt	
FeatEn		eine Codestelle sehe, an der eine Klasse viele Informationen aus einem Objekt holt um sie zu verrechnen	
ConPol		eine Kette von if-instanceof-Konstrukten sehe	
ConPol		ein switch-Statement sehe	
RefUnd		eine Klasse oder Methode sehe, deren Verantwortlichkeit oder Funktion ich nicht schnell und einfach erkenne	
SingRe		eine Klasse finde, die mehr als eine Zuständigkeit hat	
ZusDam		eine Klasse mit wenig Funktionalität sehe	
ZusWac		eine Gruppe von einfachen Daten sehe, die immer gemeinsam auftreten	
neuer Code		eine neue Klasse im Produktivcode anlege	

Merke die Ereignisse. (Nicht notieren - dies ist ein Gedächtnistraining!) Beim Eintritt des Ereignisses, handle entsprechend. Am Ende des Arbeitstages, notiere die gewählten Ereignisse und Erfolg/Mißerfolg/Nichteintreten.

Es empfiehlt sich, die Ereignisse in immer neuer Kombination auszuwählen, so daß jedes gelegentlich einmal vorkommt. Erfinde ggf. neue Ereignisse (alles ist geeignet, das zum Schreiben von mindestens einem Test veranlaßt).