Testverfahren

Ihre Einsatzgebiete und entdeckbare Fehler

Testverfahren	Mögliche Einsatzgebiete	Entdeckbare Fehler
Äquivalenzklassenbildung	• Test einzelner Variablen • Grundfunktionalitäten • Smoketest	• Fehlerhafte Wertebereiche einzelner Variablen
Grenzwertanalyse	• Randbereiche von Variablen • Smoketest • Regressionstest • Funktionaler Test	• Fehlerhafte Grenzen • Systemgrenzen falsch umgesetzt • Verschobene Grenzen
Entscheidungstabellen	• Geschäftsregeln • Entscheidungslogiken • Review von Regelsätzen des Kunden	• Falsch umgesetzte Regeln • Vergessene Regeln • "Sonderfälle"
Ursache-Wirkungs-Graph-Analyse	• Geschäftsregeln • Entscheidungslogiken • Review von Regelsätzen des Kunden • Schaltlogiken • Als Alternative zu den Entscheidungstabellen	• Falsch umgesetzte Regeln • Vergessene Regeln • "Sonderfälle"
Zustandsbasierter Test	• Automatisierungs-technik • embedded systems • Dialogbasierte Anwendungen • reaktive/sicherheits-kritische Software	• fehlerhafte und falsche Zustandsänderungen • ungültige Übergänge • falsche Reaktionen auf Eingaben
Anwendungsfallbasierter Test	• Geschäftsprozesse • Szenarien • oft im Abnahmetest	• Probleme im Prozessfluss • Schnittstellenfehler
Klassifikationsbaum & orthogonale Arrays	• Kombination mehrerer Parameter • Erstellung einer statistischen Verteilung von Kombinationen • Konfigurationstests	• Hängen von Parametern und Kombinationen ab
Wertebereichsanalyse	• Interagierende Äquivalenzklassen mit Abhängigkeiten	• Fehlerhafte Abhängigkeiten • Fehlerhafte Umsetzung von Variablenregeln

Testverfahren	Mögliche Einsatzgebiete	Entdeckbare Fehler
Testen mit User Stories	• Im agilen Bereich • ähnlich anwendungsfall-basierter Test	• Verstoß gegen Akzeptanzkriterien der User Story
Error guessing	• komplementäre Unterstützung systematischer Verfahren • nur mit erfahrenen Testern!	• "Zielfehler" aufdecken • typische Fehler der Vergangenheit finden
Exploratives Testen	• Bei instabiler Software • Smoketests • hoher Zeitdruck • schlechter/nicht vorhandener Spezifikation	• hängt vom Tester ab
Checklistenbasiertes Testen	• Testen bekannter Software • Test, um typische Funktionen zu prüfen	• hängt von der Checkliste ab
Fehlertaxonomien	• Testen auf bekannte Fehler • Nach Ermittlung von Fehlerpotentialen mittel Fehleranalysen wie z.B. FMEA, PAAG, HAZARD-Analyse, HACCP, usw.	• hängt von der Taxonomie ab • Zielfehler werden oft gut aufgedeckt
Fehlerangriffe	• Prüfung der Software auf Schwachstellen • Nutzen bekannten Softwareschwächen um Fehler zu provozieren	• hängt vom Tester ab • Unverständliche Fehlermeldungen
Anweisungsüberdeckung	• Im Komponententest auf Codeebene	• Nicht erreichbaren Code • Toten Code • Easter Eggs • Backdoors
Zweigüberdeckung	• Im Komponententest auf Codeebene	• Vergessene Anweisungen • Nicht erreichbare Zweige • Vergessene Fehlerbehandlungen
Pfadüberdeckung	• Im Komponententest auf Codeebene	• Fehlerhafte Programmpfade • Probleme bei Abhängigkeiten im Code

Testverfahren	Mögliche Einsatzgebiete	Entdeckbare Fehler
Entscheidungsüberdeckung	• Im Komponententest auf Codeebene	• Fehlerhaft ausgewertete Bedingungen • Falsch programmierte Entscheidungen
Einfache Bedingungsüberdeckung	• Im Komponententest auf Codeebene	• Keine definierten
Bedingungs-/Entscheidungs-überdeckung	• Im Komponententest auf Codeebene	• Falsch ausgewertete Bedingungen • Verwechselte Vergleichsoperatoren • Fehlerhafte logische Verknüpfungen
Minimale Mehrfachbedin-gungsüberdeckung	• Im Komponententest auf Codeebene	• Falsch ausgewertete Bedingungen • Falsche Abhängigkeiten bei atomaren Teilausdrücken
Modifizierte Bedingungs-/ Entscheidungsüberdeckung (MCDC)	• Im Komponententest auf Codeebene	• Falsche Entscheidungen, nach Bedingungsauswertung • Fehlerhaftes Verhalten einzelner Atome
Mehrfachbedingungs-überdeckung	• Im Komponententest auf Codeebene	• Alle bedingungs- und entscheidungs-bezogenen Fehler
Linear Code Sequence and Jump (LCSAJ)	• Im Komponententest auf Codeebene	• Falsche Sprungmarken • Fehlerhafte Codeflüsse • Falsche Sprungziele • Fehlerhafte Schleifen • Falsche Abbruchbedingungen in Schleifen