Fehlermanagement einsetzen

Bereich
Projektplanung und -Steuerung

Aktivität
Projekt planen

schnell durchführbar ★★☆

einfach durchführbar ★★☆

agil einsetzbar ★★★

Ziele

- Fehlerklassifikation erstellen
- Fehlerstatusmodell einführen
- Einheitliches Schema für Fehlermeldungen
- Nachvollziehbarkeit und Reproduzierbarkeit

Motivation/Problemstellung

Zur effektiven Verwaltung und Bearbeitung von Fehlermeldungen sollte ein systematisches Fehlermanagement durchgeführt werde. Tester oder weitere Stakeholder, die Fehler finden, sollten dafür eine Fehlermeldung erfassen, welche an die Entwickler zur Behebung weiter gegeben wird. Ohne ein explizites Fehlermanagement können Fehlermeldungen verloren gehen, oder in einem Zustand sein, der es der Entwicklung schwer macht, die Meldungen sinnvoll auszuwerten. Schreibt jede Person eigene Meldungen ohne konkrete Vorgaben, so ist das Risiko hoch, dass diese Informationen nicht nachzuvollziehen sind. Wenn die Meldungen nicht einheitlich in einer gemeinsamen Basis dokumentiert werden, kann kein genauer Überblick über den Gesamtzustand über z.B. offene und geschlossene Fehlermeldungen gegeben werden. Zudem ist nicht jeder Fehler gleich wichtig, das heißt, manche Meldungen müssen bevorzugt behandelt werden, es muss aber dazu erkennbar sein, welche die wichtigsten sind.

Kurzbeschreibung

Fehlermeldung sollten in einem einheitlichen Schema geschrieben werden, dass im Idealfall im Team abgesprochen ist. Ziel ist hierbei, die Nachvollziehbarkeit durch die Entwickler und die Reproduzierbarkeit der gemeldeten Fehler zu ermöglichen. Weiterhin sorgt ein Fehlerstatusmodell für einen exakt definierten Ablauf bei der Bearbeitung der Fehlermeldungen. Die Fehlerklassifikation wiederum erlaubt die Priorisierung bestimmter Meldungen je nach der Wichtigkeit / Kritikalität der Meldung.

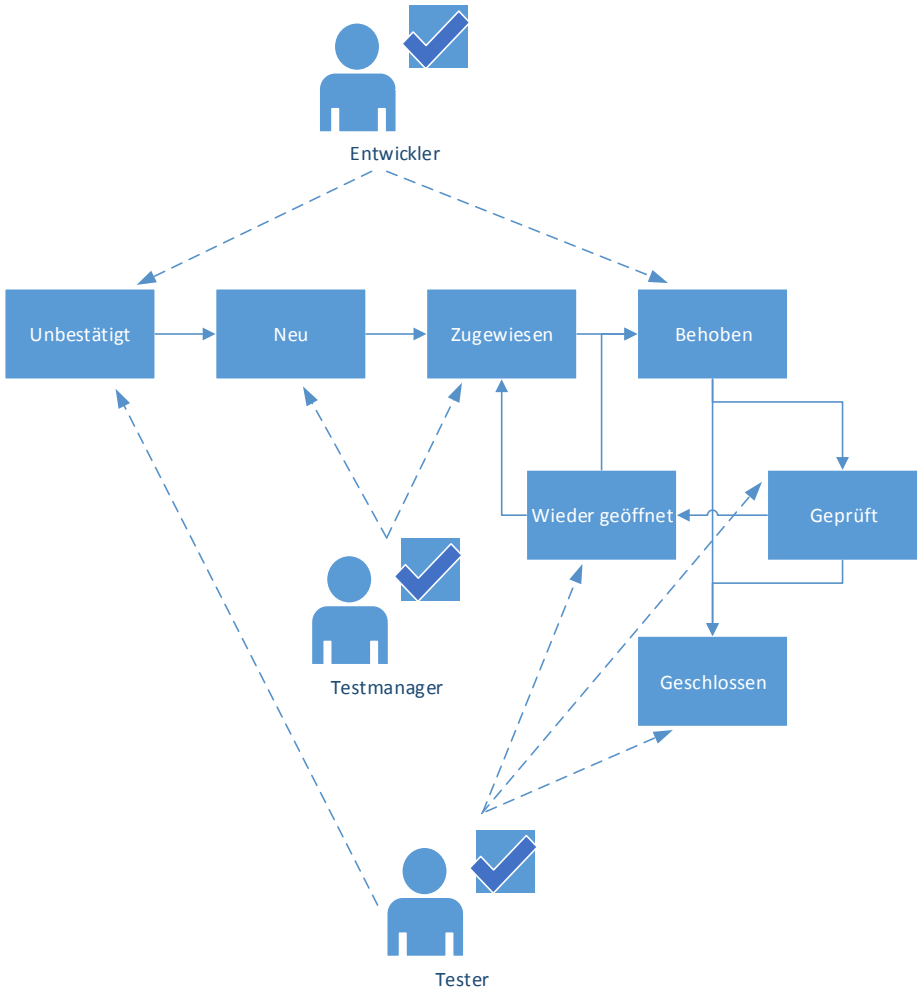

Abbildung 1: Fehlermanagement: Lebenszyklus von Fehlern und Rollen

Input

- Testfälle

Output

- Fehlermeldungen, Fehlerstatusmodell, Fehlerklassifikation.

Rahmenbedingungen

Ausführender
Tester, Testmanager

Werkzeuge, Hilfsmittel
Bug Tracking System

Vorkenntnisse/Erfahrungen
Grundkenntnisse im Testen

Ort/Umgebung
Büro

Weitere Teilnehmer
-

Voraussichtliche Dauer
Ca. 2 Stunden

Vorgehensweise

Vorbereitung
-

Durchführung
Das Fehlerstatusmodell sollte mehrere Zustände und genau definierte Zustandsübergänge haben, die jeweils von bestimmten Rollen ausgelöst werden dürfen. Ein Beispiel findet sich in Abbildung 1: jeder Entwickler oder Tester darf Fehlermeldungen öffnen (unbestätigt). Ein Dispatcher, welcher zum Beispiel ein erfahrener Entwickler oder der Testmanager sein kann, prüft jedes Ticket und akzeptiert das als neu, sofern es sich um ein Problem handelt. Nach der Bearbeitung werden die Meldungen von der Entwicklung auf behoben gesetzt. Die Tester sollten anschließend nach einer Überprüfung die Meldung schließen (geschlossen) oder zurück an die Entwicklung geben, falls der Fehler weiterhin besteht (Wieder geöffnet). Ein behobener Fehler kann alternativ auch ohne erneute Prüfung geschlossen werden. Insgesamt kann mit einem solchen Modell ein definierter Arbeitsablauf geschaffen werden, indem für jede Meldung der aktuelle Zustand und die zuständige Person jederzeit klar definiert sind.
Eine Fehlerklassifikation erlaubt die Festlegung von Reihenfolgen für die Bearbeitung von Fehlermeldungen. Denkbar sind etwa die Kategorien Kritikalität oder Schwere, Priorität, oder Häufigkeit des Auftretens. Festlegen kann die Priorität der Product Owner, bei Kategorien wir Schwere oder Häufigkeit hat oftmals der Entwickler hier das bessere Wissen. Es müssen nicht unzählige Detailabstufungen vorgenommen werden, oftmals reicht eine drei- bis fünfstufige Skala. Die Klassifikation kann auch den Bereich beinhalten, in dem der Fehler auftritt, oder die Auswirkungen (z.B. Performance Funktionalität, Security, Usability, etc.).
Beide, das Fehlerstatusmodell und die Fehlerklassifikation sollten Eingang in ein einheitliches Schema für die Fehlermeldungen finden. Dieses Schema sollte unbedingt eine genaue Beschreibung der Schritte beinhalten, die zu einem Fehlverhalten geführt haben. Dieses Verhalten muss beschrieben werden, optional mit einem Bildschirmfoto, falls möglich. In jedem Fall sollen die Nachvollziehbarkeit und die Reproduzierbarkeit durch die Entwickler gegeben sein. Dazu werden neben dem zugrunde liegenden Testfall auch die getestete Version und das Datum, gegebenenfalls auch die Testumgebung vermerkt. Im Falle von Nachfragen sollen die Verfasser der Meldungen dokumentiert werden. Selbst wenn ein Fehlverhalten nicht im Test reproduziert werden kann, sollte es gemeldet werden, mit höchstmöglicher Detaillierung, so dass die Entwicklung möglichst viele Informationen für Debugging Aktivitäten hat.

Nachbereitung/Auswertung
Nach Etablierung dieser Modelle muss auf deren Einhaltung bei der Dokumentation und Bearbeitung der Fehler geachtet werden.

Gütekriterien/Empfehlungen

Diese Best Practice sollte zu Anfang der Testaktivitäten durchgeführt werden. Die Modelle sollten mittels eines Bug Tracking Werkzeuges umgesetzt werden, das die Dokumentation der Fehlermeldungen und die Kommunikation unterstützt.
Fehlermeldungen sollten zudem regelmäßig ausgewertet werden, um die Art der Fehler zu analysieren und somit die Qualitätssicherung optimieren zu können. Zudem sollten alle Personen das gleiche System nutzen und sich an definierte Abläufe halten. Somit ist jederzeit eine einfache Übersicht zur Qualitätssicherung und der Qualität des Produkts zu gewinnen (z.B. durch Anzahl offene vs. geschlossene Fehler; Fehler in einem bestimmten Zustand, etc.).

Risiken

Die Fehlerklassifikation, die Schemata für die Meldungen und das Fehlerstatusmodell sollten nur so weit aufgebaut und detailliert werden, wie es im aktuellen Projekt notwendig ist. Ansonsten entsteht ein unnötiger Mehraufwand. Bei Einführung solcher Modelle muss darauf geachtet werden, dass sie auch konsequent genutzt werden. Sind weitere Wege für die Meldung von Fehlern etabliert, kann dies dazu führen, dass manche Meldungen ins Leere laufen, weil sich bestimmte Personen nur auf bestimmte Kommunikationskanäle konzentrieren. Generell muss bei Fehlermeldungen auf eine konstruktive und neutrale Sprache geachtet werden, da sonst Differenzen zwischen Mitarbeitern entstehen können.

Einordnung in den agilen Referenzprozess

Mögliche Vorgänger

- Testen\Systematische Testfallableitung und Tests durchführen
- Testen\Teststrategie festlegen und Teststufen aufeinander abstimmen

Mögliche Nachfolger

- Testen\Erfahrungsbasiertes Testen durchführen

Mögliche Alternativen, verwandte Praktiken

- Unterstützung\Checklisten verwenden
- Unterstützung\Standardisierte Dokumentstrukturen verwenden
- Architektur\Feature-Entwicklung und interne Qualitätsaufgaben verzahnen

Einordnung in das PQ4Agile-Qualitätsmodell

Prozesskonformität. Prozessanwendungsqualität: Effektivität, Effizienz. Prozessstabilität

Schlagworte

Fehler, Klassifikation, Qualitätssicherung

Weiterführende Informationen

Literatur

IEEE 829: IEEE Standard for Software and System Test Documentation, IEEE Computer Society, 2008.
IEEE 1044: IEEE Standard Classification for Software Anomalies, IEEE Computer Society, 2009.
A. Spillner, T. Linz: Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB-Standard, dpunkt.verlag GmbH, 2012.
A. Spillner, T. Roßner, M. Winter, T. Linz: Praxiswissen Softwaretest - Testmanagement: Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard, dpunkt.verlag GmbH, 2014.

Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im Rahmen der Maßnahme KMU-innovativ: IKT (01|S13032) gefördert.