Fragebogenkatalog zur Reifegradmessung

(Version: Fragenkatalog_Statische_Code_Analyse_v2.0.0.docx)

Anleitung:

Um die Reifegradmessung durchzuführen, müssen die folgenden Fragen beantwortet werden. Es darf für jede Frage nur eine Antwort ausgewählt bzw. angekreuzt werden.
Zu jeder Antwortmöglichkeit finden Sie die entsprechende Punktezahl in der rechten Spalte der jeweiligen Tabelle. Um die Gesamtpunkte zu bestimmen, müssen die Punkte der jeweiligen Fragen addiert werden. In der untenstehenden Tabelle kann der Reifegrad anhand der Gesamtpunkte bestimmt werden.
Aber Achtung: Liegt die Punktzahl bei einer oder mehreren Fragen unter der angestrebten Stufe, ist der Reifegrad noch nicht erreicht und es muss zunächst an diesen Stellen nachgebessert werden. Unter der Reifegradmesstabelle finden Sie die verschiedenen Verbesserungsvorschläge, die zum Erreichen des nächsten Reifegrades notwendig sind.

Statische Codeanalyse

Frage 1		
Gibt es im Projekt abgestimmte Richtlinien zur Programmierung?		
◯	Nein, jeder Entwickler bringt seinen eigenen Stil und Vorlieben ein.	0
◯	Absprachen und Richtlinien werden teamintern definiert und umgesetzt, soweit sie vom jeweiligen Entwicklungsteam als sinnvoll erachtet werden.	1
◯	Es existieren definierte formale Vorgaben und Richtlinien der Programmierung, die teamübergreifend abgestimmt wurden und obligatorisch zum Einsatz kommen. Im Rahmen von Schulungen werden diese in der Organisation verbreitet.	2
◯	Die zur Anwendung vorgegebenen Richtlinien werden fortlaufend überprüft und verbessert.	3

Frage 2		
Wird die Statische Codeanalyse als Methode im Projekt eingesetzt?		
◯	Die statische Codeanalyse wird gelegentlich freiwillig und spontan eingesetzt. Es erfolgt keine Dokumentation.	0
◯	Es kommt bei Bedarf zu geplanten Reviews von Entwicklungsergebnissen, deren Dokumentation sich von Fall zu Fall unterscheidet.	1
◯	Die Methoden der statischen Codeanalyse werden regelmäßig als Werkzeug im Entwicklungsprozess eingesetzt. Es kommt zu Sourcecode-Reviews und automatisierten Prüfungen und die Ergebnisse zur Weiterbearbeitung werden einheitlich dokumentiert.	2
◯	Es erfolgt organisationsweit eine regelmäßige Analyse und stetige Verbesserung des erreichten Standards beim Einsatz der statischen Analyse.	3

Frage 3		
Erfolgt die Bereitstellung der Software in einem definierten Workflow (z.B. CI/CD Pipeline), in den die statische Quellcodeanalyse eingebettet werden kann?		
◯	Nein, es gibt keinen solchen Ansatz.	0
◯	Es gibt automatisierte statische Prüfungen sowie Sourcecode-Reviews, die vor der Bereitstellung der Software ausgeführt und dokumentiert werden.	1
◯	Es erfolgt die automatisierte Integration der statischen Methoden in einen CI/CD Workflow. Dieser ist auf die Bedürfnisse des Projekts zugeschnitten und konfiguriert.	2
◯	Die statischen Methoden sind fest im Workflow der CI/CD Pipeline installiert. Die Analyseergebnisse werden benutzt, um die Codequalität kontinuierlich zu verbessern.	3

Frage 4		
Wie erfolgt die Durchführung einer statischen Prüfung?		
◯	Die sporadischen statischen Prüfungen des Quellcodes erfolgen ausschließlich durch Sourcecode-Reviews.	0
◯	Sourcecode-Reviews sind der wichtigste Bestandteil der statischen Prüfungen. Diese werden durch automatisierte Prüfungen (innerhalb der Entwicklungsumgebungen oder durch andere Werkzeuge) ergänzt.	1
◯	Zum Einsatz kommt eine Kombination aus Sourcecode-Reviews und automatisierten statischen Analysen, um die jeweiligen Vorteile der Methoden bestmöglich zu nutzen und deren Ergebnisse als Eingabe für andere Testverfahren zu nutzen.	2
◯	Der Einsatz von Sourcecode-Reviews und automatisierten Verfahren im Entwicklungsprozess ist etabliert und wird durch ständige Fort- und Weiterbildung des Teams gefördert und gegebenenfalls weiterentwickelt.	3

Frage 5		
Finden regelmäßig strukturierte Tests zur Prüfung der behobenen Abweichungen im Nachgang zur statischen Analyse statt?		
◯	Es erfolgen spontane Tests der manuell durchgeführten statischen Analysen.	0
◯	Zusätzlich zu den bereits geplanten Tests werden spezielle Tests erstellt. Diese sollen Schwachstellen und korrigierte Abweichungen prüfen, die durch die statischen Analysen entdeckt wurden. Die Testergebnisse werden nach Projektvorgaben dokumentiert.	1
◯	Die statischen Analysen werden fester Bestandteil der definierten Teststrategien. Die Tests der korrigierten Abweichungen aus diesen Bereichen werden in der Regressionsstrategie berücksichtigt. Eine Dokumentation erfolgt gemäß den organisationsweiten Vorgaben.	2
◯	Die Tests der Abweichungen auf Basis der statischen Analysen werden in den Entwicklungsprozess integriert und projektübergreifend optimiert.	3

Frage 6		
Wird das Prinzip des Kontinuierlicher Verbesserungsprozesses verfolgt?		
◯	Das Prinzip des KVP wird nicht umgesetzt.	0
◯	Der Kontinuierlicher Verbesserungsprozess wird durch stetige Plananpassung umgesetzt.	1
◯	Der Kontinuierliche Verbesserungsprozess bezieht die Ressourcensituation mit ein (Geld, Zeit, Personal).	2
◯	Regelmäßige Reviews und Retrospektiven sowie die ständige Fort- und Weiterbildung aller Mitarbeitenden stellen sicher, dass der KVP gelebt wird.	3

Punktestand: ☐

„Reifegradmesstabelle“:

Punkte	Reifegrad
0 – 5	Initial
6 – 11	Kontrolliert
12 – 17	Effizient
18	Optimierend

Verbesserungsvorschläge:

von „Initial“ zu „Kontrolliert“

- Programmierrichtlinien definieren, sich dabei an gängigen Standards (z.B. SUN Java Code Conventions) orientieren
- Verbindliche Umsetzung der Programmierrichtlinien in der Entwicklungsumgebung und Anwendung bereits während der Programmierung.
- Regelmäßige Code Reviews einführen
- Automatische Sourcecode-Analyse durch Einsatz geeigneter Werkzeuge wie PMD oder findbugs etablieren
- Ergebnisse der Analysen dokumentieren

von „Kontrolliert“ zu „Effizient“

- Schulungen zur Verbesserung der handwerklichen Qualität des Source Code anbieten
- Echtzeit-Codeanalyse umsetzen (z.B. mit SonarQube)
- Analyseergebnisse, insbesondere zu Komplexitätsgraden, in eine Risikobasierte Teststrategie integrieren

von „Effizient“ zu „Optimierend“

- Regelmäßige Schulungen des gesamten Teams unterstützen die Einführung von Standards.
- Regelmäßige Prüfung der eingesetzten Werkzeuge auf mögliche und notwendige Erweiterbarkeit vornehmen
- Die Qualität der Korrekturmaßnahmen regelmäßig verfolgen, überprüfen und verbessern