

Reviews von Entwicklungsartefakten durchführen

Bereich
Evaluation

Aktivität
Reviews durchführen

Ziele

- Fehler und Probleme frühzeitig finden
- Wissenstransfer ermöglichen
- Teamzusammenhalt fördern
- Lösungen erarbeiten

schnell durchführbar ★★☆☆

einfach durchführbar ★★☆☆

agil einsetzbar ★★☆☆

Motivation/Problemstellung

Je früher Fehler gefunden werden, desto günstiger ist deren Behebung. Werden beispielsweise in einem Systemtest Fehler gefunden, die auf inkorrekten Anforderungen basieren und somit viel früher zu finden gewesen wären, so müssen unter Umständen mehrere Stufen im Entwicklungsprozess wiederholt werden. Mit Reviews (z.T. auch statische Tests oder Inspektionen genannt) können Artefakte, die im Entwicklungsprozess erstellt werden, direkt nach der Erstellung geprüft werden, so z.B. Anforderungen, Lösungskonzepte, Code, oder auch Testfälle. Insbesondere möchte man kritische Fehler mit Reviews finden, also solche Fehler, die erhebliche Auswirkungen haben können (z.B. den Betrieb der Applikation gefährden), damit sich Reviews lohnen. Neben der Fehlerfindung, welches ein wesentliches Ziel von Reviews darstellt, tragen Reviews auch zu Wissenstransfer bei, insbesondere wenn Gruppenaktivitäten Teil des Reviewprozesses sind. In diesem Fall kann auch der Teamzusammenhalt gefördert werden. Je nach Ausrichtung des Reviews kann auch konstruktiv nach Lösungen gesucht werden (typischerweise spricht man hier von einem Walkthrough).

Kurzbeschreibung

Nachdem ein Entwicklungsartefakt (z.B. Anforderungen, Design, Code) erstellt wurde, kann es von einem oder mehreren Reviewern überprüft werden. Der bzw. die Reviewer können das entsprechende Dokument basierend auf ihren eigenen Erfahrungen nach Fehlern prüfen, sowie Anmerkungen und Verbesserungsvorschläge dokumentieren (direkt in dem Dokument, in einer Fehlerliste, oder werkzeugunterstützt). Darüber hinaus besteht die Möglichkeit, den Reviewern Leseunterstützung bei der Prüfung zur Verfügung zu stellen, beispielsweise in Form einer Checkliste, die eine Anzahl von Fragen enthält, auf die das Dokument zu prüfen ist. Neben der Möglichkeit, die gleiche Checkliste für alle Reviewer zu nutzen, können auch unterschiedliche Checklisten, die verschiedene Arten von Fehlern bzw. Perspektiven abdecken, genutzt werden. Darüber hinaus können auch Szenarien für ein Review eingesetzt werden, welche den Reviewer aktiv auffordern, eine Aktivität durchzuführen, und dabei die Qualität zu prüfen. Beispielsweise kann ein Testerszenario einen Reviewer auffordern, Testfälle aus einem Anforderungsdokument oder auch User Stories abzuleiten, und dabei zu prüfen, ob alle Informationen vorhanden sind.

Im Anschluss an das Review können die Auffälligkeiten dem Autor in einer Gruppensitzung kommuniziert bzw. direkt übergeben werden. Der Autor ist dann schließlich für die Korrekturen verantwortlich.

Weitere Detailbeschreibungen sind auch in den Best Practices „Usability-Review durchführen“ sowie „Anforderungen reviewen“ zu finden.

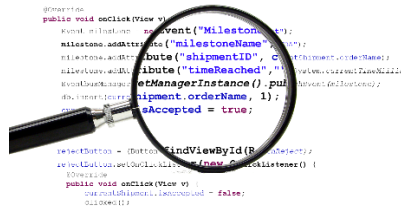


Abbildung 1: Review eines Entwicklungsartefaktes

Input

- Das bzw. die zu prüfenden Artefakte, optional Leseunterstützung wie Checklisten, optional Template zur Dokumentation von Auffälligkeiten

Output

- Dokumentation der Auffälligkeiten, korrigiertes Artefakt

Rahmenbedingungen

Ausführender

Reviewer, Tester, Entwickler

Werkzeuge, Hilfsmittel

Werkzeuge zur Unterstützung der Reviews, Leseunterstützung

Vorkenntnisse/Erfahrungen

Entwicklung, Test, QS

Ort/Umgebung

Büro, Meetingraum

Weitere Teilnehmer

Je nach Bedarf weitere Experten aus dem Entwicklungsprozess

Voraussichtliche Dauer

Je nach Größe des zu untersuchenden Artefaktes und Fokus des Reviews, in der Regel zwischen 30 Minuten und max. 4 Stunden

Vorgehensweise

Vorbereitung

Häufig gibt es einen Organisator für das Review, welcher auch der Autor des zu reviewenden Dokuments sein kann. Es sind die Rollen zu bestimmen, die im Review eingenommen werden sollen und, sofern existierend, die jeweiligen Checklisten und die Aufgaben zu erstellen. Die Checklisten selbst können zwar anhand von Vorlagen erstellt werden, sind aber dennoch an den eigenen Kontext und die eigenen Bedürfnisse anzupassen. Ebenso sollen die durchführenden Personen bestimmt und ggf. Räume reserviert werden. Falls ein Werkzeug unterstützend eingesetzt werden soll, beispielsweise für eine statische Prüfung des Codes, kann dies im Vorfeld des Reviews durchgeführt werden, es ist jedoch in der Regel nicht Teil des Reviewprozesses.

Durchführung

Zunächst ist zu entscheiden, ob ein optionales Überblickstreffen sinnvoll ist. In einem solchen kurzen Treffen stellt der Autor das zu prüfende Dokument überblicksmäßig vor, kann den Fokus für die Reviewer noch einmal darstellen, und für Fragen zur Verfügung stehen. Typischerweise wird ein solches Treffen dann durchgeführt, wenn neue Reviewer eingebunden sind, die Reviewer das Dokument nicht kennen, oder komplexe Bereiche im Dokument existieren, die viel eigenen Einarbeitungsaufwand erfordern würden.

Im Anschluss startet das eigentliche Review, welches zunächst individuell von jedem Reviewer durchgeführt wird. In der Regel ist das primäre Ziel bei der Durchführung eines Reviews das Finden von Auffälligkeiten und Fehlern. Erfahrene Inspektoren benötigen häufig keine Leseunterstützung, sondern finden Probleme, Auffälligkeiten und Fehlern aufgrund ihres Wissens. Bei der Nutzung von Checklisten konzentrieren sich die Reviewer auf die aufgeführten Fragen und prüfen aus dieser Perspektive die ihnen zugewiesenen Artefakte. Die gefundenen Fehler werden dokumentiert. Auffälligkeiten können dabei direkt nach der vermuteten Schwere oder Kritikalität bewertet werden, um die spätere Korrektur zu priorisieren. Um den Aufwand für spätere Schritte minimal zu halten, empfiehlt sich auch ein direktes Markieren und Dokumentieren im zu untersuchenden Artefakt selbst.

Im Anschluss an die individuelle Fehlersuche kann ein gemeinsames Meeting durchgeführt werden, in dem die gefundenen Auffälligkeiten gesammelt werden. Zudem können hier auch Fragen geklärt, sowie Verbesserungsvorschläge aufgenommen werden. Es ist von einem Moderator darauf zu achten, dass ein solches Treffen nicht zu diskussionsintensiv wird – es dient primär nicht dazu, Lösungen konstruktiv zu erarbeiten. In einem solchen Meeting kann, neben dem Zusammentragen von Fehlern, auch gemeinsam nach neuen Fehler gesucht werden, dies ist jedoch im Allgemeinen nur für besonders kritische Bereiche sinnvoll und effektiv. Resultat des Meetings ist eine Liste von Fehlern und Verbesserungsvorschlägen. Ein solches Treffen kann, neben der Fehlerliste auch dazu beitragen, Wissen zu streuen bzw. das Teambuilding zu fördern. Der Moderator muss dazu unbedingt sicherstellen, dass stets das Dokument bewertet wird, und nicht der Autor, also dass gemeinsam eine hohe Qualität des zu untersuchenden Artefakts sichergestellt werden möchte. Alternativ zu einem gemeinsamen Treffen können Auffälligkeiten direkt an den Autor geschickt werden; in diesem Fall muss allerdings der Autor alle Notizen alleine sichten und ggf. Nachfragen an die jeweiligen Reviewer stellen. Ein Gruppenlernerfolg bleibt zudem bei diesem Vorgehen aus.

Sollte es Bedarf an gemeinsamen Diskussionen zu einzelnen Auffälligkeiten geben bzw. an dem Suchen konstruktiver Lösungen, so ist ein separates Treffen zu planen und durchzuführen (eine sogenannte „third hour“).

Nachbereitung/Auswertung

Die in einem Meeting erarbeitete und konsolidierte Fehlerliste nutzt der Autor, um Korrekturen durchzuführen; er entscheidet letztlich, was und wie er gefundene Fehler korrigiert, und ist dafür auch verantwortlich. Als finales Qualitätstor kann sich der Autor nach der Korrektur noch einmal mit dem Organisator des Reviews, dem Moderator, oder einer anderen Person zusammen setzen, um die Korrekturen zu besprechen, und somit auch Folgefehler möglichst gering zu halten.

Gütekriterien/Empfehlungen

Im agilen Umfeld sind primär leichtgewichtige Reviews zu empfehlen, d.h., geringe Planung, keine langen Meetings, überschaubare Mengen an Dokumenten für die

Reviews, leichtgewichtige Dokumentation (also z.B. direkt im Dokument). Bei kritischen Anforderungen oder zum Wissensaustausch sind Meetings nach der Fehlersuche auch im agilen Umfeld durchaus sinnvoll. Checklisten unterstützen gerade noch nicht so erfahrene Reviewer. Weiterhin ist es sinnvoll, einen Reviewverantwortlichen zu benennen, der die Pflege des Reviewprozesses und die Pflege von Checklisten sicherstellt, sowie für Fragen rund um das Thema zur Verfügung steht. Reviews gelten, sofern sie zielgerichtet durchgeführt werden, als mit zu den effizienten Qualitätssicherungsmethoden, auch im agilen Umfeld. Für unterschiedliche Dokumente können durchaus unterschiedliche Reviewprozesse gewählt werden (z.B. mehr oder weniger systematisch, mehr oder weniger involvierte Personen, mehr oder weniger eingesetzter Aufwand), und nicht alle Entwicklungsartefakte bzw. Bereiche in Artefakten müssen notwendigerweise gereviewt werden. Es sollte aus unterschiedlichen Blickwinkeln auf Dokumente geschaut werden, je nach Bedarf an die jeweiligen Dokumente. Beispiele sind „Tester“ für das Review der Anforderungen und die dazugehörige Aufgabe „Test ableiten“, um frühzeitig zu prüfen, ob die Anforderungen für einen Systemtest gebräuchlich sind. Im agilen Umfeld ist es beispielsweise sinnvoll, dass jede (komplexe, kritische) Userstory aus der Perspektive eines Entwicklers und Testers gereviewt wird und Auffälligkeiten direkt geklärt werden. Zuletzt soll erwähnt werden, dass Pair Programming durchaus als eine Reviewtechnik angesehen werden kann und auf Codeebene eine sinnvoll ergänzende Qualitätssicherungsmaßnahme für Teilbereiche darstellen kann.

Risiken

Reviews dürfen nicht zur Überprüfung und Bewertung einzelner Personen verwendet werden, sondern zur Verbesserung von Entwicklungsartefakten. In Checklisten soll lediglich eine kleine Menge an Fragen (typischerweise 3-7) enthalten sein, um das Review nicht zu überfrachten. Exemplarische Checklisten findet man unter www.software-reviews.de/checklisten.php oder <http://inspection.iесе.de>. Reviews sollten geplant werden, insbesondere soll ausreichend Zeit für das zu reviewende Dokument eingeplant werden, damit das Review auch effektiv ist.

Einordnung in den agilen Referenzprozess

Mögliche Vorgänger

- Testen\Systematische Testfallableitung und Tests durchführen
- Testen\Teststrategie festlegen und Teststufen aufeinander abstimmen
- Architektur\Architekturlösungen im Team entwickeln
- Architektur\Grob- und Detailplanung bei der Implementierung nutzen
- Architektur\Kontinuierliche Architekturbewertung durchführen
- Architektur\Architekturentscheidungen dokumentieren
- Architekturrelevante Anforderungen dokumentieren

Mögliche Nachfolger

-

Mögliche Alternativen, verwandte Praktiken

- Unterstützung\Checklisten verwenden
- Unterstützung\Standardisierte Dokumentstrukturen verwenden

Einordnung in das PQ4Agile-Qualitätsmodell

Produktqualität: Funktionale Tauglichkeit, Zuverlässigkeit, Sicherheit, Wartbarkeit, Übertragbarkeit. Strukturqualität: Fähigkeit und Schulung der Personals, Bewusstsein des Personals

Schlagworte

Statischer Test, Review, Inspektion, Qualitätssicherung

Weiterführende Informationen

Literatur

D. Masak: Der Architekturreview: Vorgehensweise, Konzepte und Praktiken, Springer Xpert.press, 2010.

K. Wiegers: Peer Reviews in Software, A Practical Guide. Addison-Wesley, 2002.

Best Practice „Reviews von Entwicklungsartefakten durchführen“
Version 1.0 – 15.06.2015 – Autor: Frank Elberzhager, Alexander Klaus, Fraunhofer IESE
Das Projekt PQ4Agile wird vom Bundesministerium für Bildung und Forschung im
Rahmen der Maßnahme KMU-innovativ: IKT (01 | S13032) gefördert.