

Basis-Testkonzept

Für Software-Entwicklungsteams



Einleitung

Meinen aktuellen Erfahrungsstand im Hinblick auf eine pragmatische Qualitätssicherung möchte ich mit diesem Dokument gratis zur Verfügung stellen, um einen Beitrag dazu zu leisten, Software insgesamt besser zu machen.

Zielgruppe

Das Konzept richtet sich an Entwicklungsteams die aktuell keine strukturierte Software-Qualitätssicherung betreiben.

Einführung

Das Konzept ist so gestaltet, dass es ohne Betreuung durch Fachpersonal eingeführt werden kann.

Eine strukturierte Begleitung sorgt dabei natürlich für schnellere Erfolge – Dabei unterstütze ich gern.

Inhalt

Zielsetzung	4
Quality Gates	5
Metriken & Dokumentation	6
Rollout Plan	7
Perspektivische Weiterentwicklung	8
Kontakt & Abschließendes	9

Über dieses Dokument

Dieses Konzept ist eine Diskussionsgrundlage und sollte nicht einfach so in der aktuellen Entwurfsform per Anordnung eingeführt werden.

Es stellt viel mehr einen ersten Wurf dar, der Vorschläge und ein konkretes Umsetzungskonzept mitbringt.

Die Zielsetzung und die Maßnahmen, dies zu erreichen, sollte von jedem Team mit seinen individuellen Charakteren selbst erarbeitet und fixiert werden.

Dafür bietet das Testkonzept Platz.

In der Folge kann es an einem zentralen Ort abgelegt werden – Viel wichtiger als die Ablage ist jedoch, dass es im täglichen Arbeiten auch gelebt wird.

Zielsetzung

Jedes Teammitglied hat einen eigenen Begriff und eine eigene Gewichtung, wenn es darum geht, die Wertigkeit eines Software-Produkts für sich persönlich zu beurteilen.

Diese Vorstellungen können in einem gemeinsamen Mission-Statement vereint werden. Wichtig dabei: Dieses Statement müsste von jedem Teammitglied so unterschrieben werden können.

Quality Gates

Die Gates konkretisieren das Ziel auf Arbeitsebene, sodass es konkret umgesetzt werden kann.

Jedes Quality-Gate sollte dabei ein Aussagesatz sein: Ist er erfüllt, gilt auch das jeweilige Quality-Gate als erfüllt. Weiter unten habe ich zur Veranschaulichung Beispiele hinterlegt.

Auch die Quality Gates müssen angepasst werden, damit sie die aktuellen Bedürfnisse des Teams widerspiegeln. Bei einer graduellen Einführung (zum Beispiel von Test-Coverage) empfiehlt es sich, das Testkonzept mit einer Zielsetzung zu versehen und regelmäßig anzupassen.

Zielsetzung

Dieses Testkonzept dient der Strukturierung der Qualitätssichernden Maßnahmen im Entwicklungsteam <Name des Teams>.

Gemeinsam sind wir am <Datum> zusammengetreten und haben uns auf den folgenden Qualitätsanspruch geeinigt:

<Qualitätsanspruch>

Beispiel: „Als Team Start Page sind wir der erste Anlaufpunkt unserer potenziellen Kund:innen. Wir legen daher einen hohen Fokus auf Optisches Gefallen und schnelle Antwortzeit, um sicherzustellen, dass wir als Unternehmen mit hochwertigen Lösungen verlässlich gute Qualität liefern – Egal von welchem Gerät aus der Zugriff erfolgt. Den Qualitätsanspruch realisieren wir in Ermangelung einer QA-Person gemeinsam und planen dafür notwendige Aufwände in den Sprint mit ein.“

Im Folgenden legen wir dar, wie wir diesen Qualitätsanspruch erreichen möchten.

(Anmerkung: Der hier bereitgestellte Beispielttext ist ein generisch. Wichtig ist es hier, einen in Meetings zitierfähigen Grundsatz zu formulieren, auf den sich bei Einwirkung von Zeitdruck und dringenden Anfragen jede:r verbindlich berufen kann.)



Maximilian Schluer GmbH
Vielohweg 119
D-22455 Hamburg
www.maximilian-schluer.de

kontakt@maximilian-schluer.de
+49 40 / 3345 3660

Amtsgericht Hamburg
HRB 170453

USt. ID: DE345247357



Quality Gates

Die folgenden Quality-Gates gelten in unserem Entwicklungsprozess:

Vor der Entwicklung

- > **Format:** Die Spezifikation folgt dem vorgegebenen Template.
- > **Akzeptanzkriterien:** Jede Anforderung enthält 3-5 Akzeptanzkriterien.
- > **Refinement:** Das Ticket wurde gemeinsam besprochen.

Während der Entwicklung

- > **Unit-Tests:** Die Unit-Tests sind grün.
- > **Coverage:** Die Testabdeckung ist bei 80 %.
- > **Statische Analyse:** Linter laufen erfolgreich.
- > **Smoke-Test:** Der abgeschlossene Change ist selbst geprüft.

Nach der Entwicklung

- > **Code-Review:** Der Code wurde von einer weiteren Person abgenommen.
- > **Abnahmetest:** Eine dritte Person hat den Change auf Funktionalität geprüft.
- > **PM-Abnahme:** Das Produktmanagement akzeptiert den Change.

Vor dem Deployment

- > **Explorativer Regressionstest:** Das gesamte Team hat die Änderungen geprüft.
- > **Stakeholder-Freigabe:** Die Änderungen wurden intern vorgestellt.

Metriken & Dokumentation

Die folgenden Dokumentationsartefakte und Metriken nutzen wir:

Metriken

- > **Unit Test Coverage** - Ziel: > 80%
- > **System Test Coverage** - Ziel: > 50%
- > **Tests** - Ziel: Status Grün; Auskommentierte Tests nur mit Bug-Tickets.
- > **Bugs** - Ziel: Keine mit Priorität > „Hoch“ offen.

Dokumentationsartefakte

- > **Tickets** - Erfasst in <System>.
- > **Lint-Konfiguration** - Erfasst in <System>.
- > **Test-Reports** - Erfasst & Archiviert in CI.
- > **Code-Review-Freigabe** - Erfasst in <System>.
- > **PM-Freigabe** - Erfasst in <System>.
- > **Abnahmetest-Report** - Erfasst als Kommentar in <System>.
- > **Stakeholder-Freigabe** - Erfasst in <System>.

(Anmerkung: Dieser relativ weitreichende Ansatz sorgt beispielsweise dafür, dass auch im Rahmen eines hohen Formalisierungsgrads die meisten Anforderungen erfüllt sind. Ausgenommen hiervon sind stark regulierte Kontexte - Hierfür sind gegebenenfalls weitere Maßnahmen zu ergreifen, um Dokumentation anzufertigen.)

Rollout Plan

Für den Rollout haben wir uns zunächst auf die folgenden Schritte geeinigt, wobei bei der Einführung neuer Schritte die Möglichkeit besteht, die zuletzt Eingeführten zu hinterfragen und anzupassen. Jede Iteration läuft <Wert> Wochen.

Jede neu eingeführte Maßnahme probieren wir für mindestens zwei Wochen aus, um Effekt und Ergonomie auf uns wirken zu lassen.

Erste Iteration

- > Einführung von statischer Code-Analyse, Erhöhung der Abdeckung wöchentlich um <Wert> Prozent bis zur Zielerreichung. Aktivierung weiterer Checks nach Möglichkeit.
- > Einführung von Monitoring der Unit Test Coverage, Erhöhung der Abdeckung wöchentlich um <Wert> Prozent bis zur Zielerreichung.
- > Jeder Change wird von nun an von der Entwickler:in getestet.

Zweite Iteration

- > Ausrollen einer Formatvorlage für Anforderungen (User Story / Summary, Acceptance Criteria, Additional Information).
- > Einführung eines Refinement-Meetings.
- > Einführung einer PO-Abnahme.

Dritte Iteration

- > Prüfung der Akzeptanzkriterien vor dem Refinement.
- > Einführung des Abnahmetests von PM & einer dritten Person, die weder an Entwicklung, noch am Code-Review beteiligt war.
- > Einladung relevanter Stakeholder:innen zur Freigabe.

(Anmerkung: Die Reihenfolge kann beliebig angepasst werden.)



Maximilian Schluer GmbH
Vielohweg 119
D-22455 Hamburg
www.maximilian-schluer.de

kontakt@maximilian-schluer.de
+49 40 / 3345 3660

Amtsgericht Hamburg
HRB 170453

USt. ID: DE345247357

Perspektivische Weiterentwicklung

Je nach Kapazität & Notwendigkeit können beispielsweise die folgenden, weiteren Maßnahmen eingeführt werden:

- > Formaleres Stakeholder-Management.
- > User-Feedback-Meetings.
- > Regelmäßige Tests durch Personal außerhalb des Teams.
- > A/B-Testing.
- > Formeller Regressionstest mit Test Case Management-System.
- > Langzeitmonitoring der Performance der Akzeptanz-Test-Runs.
- > Regelmäßige externe Akzeptanz-Tests.
- > Externe Penetration-Tests.
- > End User Tests.
- > Internes Support-Postfach für Sales & Customer Success.

(Anmerkung: Diese Liste erhebt keinen Anspruch auf Vollständigkeit.

Jede Maßnahme kommt nicht nur mit Vorteilen, sondern auch mit Risiken, die genau abgewogen werden sollten. Für eine pragmatische Qualitätssicherung gilt: Je leichtgewichtiger sie ist, desto mehr Akzeptanz erhält sie, desto unkomplizierter wird sie in der Täglichen Arbeit gelebt.)

Kontakt & Abschließendes

Wie bereits eher in diesem Dokument erwähnt, handelt es sich bei diesem Testkonzept keinesfalls um einen One-Size-Fits-All-Ansatz.

Im Gegenteil rate ich ausdrücklich dazu, Quality-Gates und die dazugehörigen Maßnahmen vom Team selbst festzulegen. Eine angeordnete Einführung kann zu erheblichen Spannungen führen, die vermeidbar sind.

Bei der Einführung sollte darauf geachtet werden, das Team nicht mit zu vielen, gleichzeitigen Änderungen zu überfordern.

Eine bis drei Anpassungen alle zwei bis vier Wochen ist eine gute Faustregel – Es sollte dabei aber auch die Möglichkeit geschaffen werden, die bisher eingeführten Regeln zu evaluieren.

Es empfiehlt sich, dies in offener Runde zu besprechen. Auch eine Rücknahme einer unpassenden Regel kann ein Erfolg sein.

Grundsätzlich auf den Prüfstand gestellt werden sollte das Konzept jeweils bei Beitritt und Verlassen eines Teammitglieds, weil jede Änderung im Gefüge möglicherweise auch eine Änderung im Qualitätsanspruch mit sich bringt.

Ist spontan keine professionelle Begleitung innerhalb des Unternehmens möglich?

Bei Einführung und Anpassung helfe ich gern:

Kontaktaufnahme per Mail an anfrage@maximilian-schluer.de.



Maximilian Schluer GmbH
Vielohweg 119
D-22455 Hamburg
www.maximilian-schluer.de

kontakt@maximilian-schluer.de
+49 40 / 3345 3660

Amtsgericht Hamburg
HRB 170453

USt. ID: DE345247357

