

Zu Qualitätssicherung

Inhalt

1	Qualitätssicherung	2
1.1	Testarten	2
1.2	Testphasen.....	3
1.3	Testphasen x Testarten.....	3
1.4	Fehlerklassifikation (nach Fehlerfolgeschwere).....	4
1.5	Bug Fixing	4
2	Testkonzept	5
2.1	Vorgehensweise	5
2.2	Testentwurfsspezifikation	5
2.3	Testfallspezifikation	5
2.4	Testablaufspezifikation	5
2.5	Testdurchführung	5
2.6	Testauswertung.....	5

Sie führen einen Test durch, der die Zusammenarbeit einiger voneinander abhängiger Komponenten Ihres Programms testet.

Um welche der folgenden Testarten handelt es sich dabei?

Tragen Sie die Ziffer vor der zutreffenden Testart in das Kästchen ein.

- ☐ 1 Modultest
- ☐ 2 Integrationstest
- ☐ 3 Akzeptanztest
- ☐ 4 Systemtest
- ☐ 5 Komponententest

1 Qualitätssicherung

Zu Qualitätsminderung einer Anwendung können einerseits fehlerhafte, fehlende oder nicht entdeckte Anforderungen des Auftragsgebers und andererseits fehlende oder fehlerhafte Implementierungen der vorhandenen Anforderungen führen.

Um dies zu vermeiden, werden während und nach der Implementierung unterschiedliche Tests durchgeführt.

1.1 Testarten

Im Allgemeinen unterscheidet man zwischen folgenden Tests.

Unit-Tests

Unit-Tests werden bereits während des Entwicklungsprozesses implementiert und durchgeführt. Sie bestehen aus dem Testen einzelner Methoden und Funktionen der von der Software verwendeten Klassen, Komponenten oder Module. Unit-Tests dienen vor allem zu Überprüfung, dass man durch Veränderungen im Code das bereits korrekt Funktionierende nicht beschädigt hat.

Funktionale Tests

Funktionale Tests konzentrieren sich auf die Anforderungen einer Anwendung. Sie überprüfen nur die Ausgabe einer Aktion und prüfen nicht die Zwischenzustände des Systems bei der Ausführung dieser Aktion.

Smoke-Tests

Mit Smoke-Tests werden schnell(!) die grundlegenden Funktionalitäten einer Anwendung geprüft

Integrationstests

Integrationstests überprüfen, ob die verschiedenen Bestandteile einer Anwendung korrekt zusammenarbeiten. Zum Beispiel wird die Interaktion mit der Datenbank getestet.

End-to-end tests

End-to-End-Tests replizieren das Benutzerverhalten mit der Anwendung in einer vollständigen Anwendungsumgebung. Dabei wird überprüft, ob verschiedene Benutzerabläufe wie erwartet funktionieren.

Für diese Tests ist eine vollständige und präzise Dokumentation der Use Cases, z.B. in Form von Storyboards unabdingbar!

Abnahmetests

Abnahmetests sind formale Tests, die überprüfen, ob ein System die Anforderungen erfüllt. Diese Tests bilden das Verhalten des Endnutzers.

Leistungstests

Leistungstests bewerten die Leistung eines Systems unter einer bestimmten Arbeitslast. Diese Tests helfen, die Zuverlässigkeit, Geschwindigkeit, Skalierbarkeit und Reaktionsfähigkeit einer Anwendung zu messen.

1.2 Testphasen

Tests werden in jeder Phase des Projektes durchgeführt. Man kann die Phasen in drei Bereiche unterteilen:

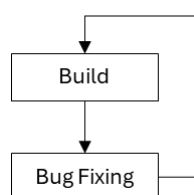
Entwicklung Während die Anwendung entwickelt wird.
In dieser Phase sind die Entwickler auch die Tester.

Build In dieser Phase gilt Entwicklungsstopp, d.h.
der Code der Anwendung darf nicht mehr geändert werden.

Die Anwendung wird zusammengeführt und alle Unit Tests werden durchgeführt.

Anschließend wird die Anwendung von den Entwicklern selbst getestet.
Dabei testet ein Entwickler nicht den von ihm implementierten Teil der Anwendung, sondern Teile, die von den Kollegen implementiert wurden.

Alle gefundenen Fehler werden dokumentiert und klassifiziert.
In dieser Phase kann auch ein Testteam die Anwendung testen.
Das Testteam hält Ergebnisse aller Tests in einem Testprotokoll fest.



Einer Build-Testphase folgt immer eine Fehlerbehebungsphase (**Bug Fixing**).

Release Die Anwendung steht bereit, dem Auftraggeber ausgeliefert zu werden.
Für Gewöhnlich testet in dieser Phase ein spezielles Tester-Team.

1.3 Testphasen x Testarten

Tests	Phase	Entwicklung	Build	Release
Unit-Tests				
Funktionale				
Smoke-Tests				
Integrationstests				
End-to-end tests				
Abnahmetests				
Leistungstests				

1.4 Fehlerklassifikation (nach Fehlerfolgeschwere)

Die Fehlerschwere wird für gewöhnlich in vier Klassen unterteilt:

- Niedrig:** Kleinere Fehler oder Mängel, die das Arbeiten mit der Anwendung nicht wesentlich behindern.
Die Behebung **können** in der nächsten Projektphase.
- Mittel:** Fehler oder Mängel deren Auswirkungen auf das auszuliefernde Phasenergebnis beschränkt sind.
Diese Fehler oder Mängel **sollten** vor der Auslieferung des Produktes behoben werden.
- Hoch:** Schwerwiegende Fehler oder Mängel, die die Konsistenz und die Qualität des auszuliefernden Ergebnisses gefährden.
Fehlen oder falsche Umsetzung der geplanten Funktionalitäten oder Teilen davon.
Diese Fehler oder Mängel **müssen** behoben werden.
- Show Stopper:** Sind Fehler, die zu Absturz der Anwendung führen.
Ihre Behebung ist **zwingend erforderlich** und muss als erstes vorgenommen werden.

1.5 Bug Fixing

Nach einer Testphase findet eine Fehlerbehebungsphase statt.

In dieser Phase werden Show-Stopper, die Fehler mit der Fehlerschwere hoch und für gewöhnlich auch mittelschwere Fehler behoben.

Diese Phase dauert für gewöhnlich ein bis zwei Tage. Anschließend werden entsprechende Teile der Anwendung wieder getestet.

Bug Fixing und anschließende Tests werden so lange durchgeführt, bis mindestens keine Show-Stopper oder Fehler mit der Fehlerschwere hoch gefunden werden.

Somit ist Bug Fixing ein iterativer Prozess und kann unter Umständen zur Verschiebung eines Release als Konsequenz haben.

2 Testkonzept

2.1 Vorgehensweise

Hier werden festgelegt

- Die Zeit für die Testphase
- Welcher Bereich der Anwendung bzw. welche Testaufgabe von wem getestet wird
- Wie genau getestet wird
- Wie die Testergebnisse protokolliert werden.

2.2 Testentwurfsspezifikation

Die Testentwurfsspezifikation

- verfeinert die Beschreibung der Vorgehensweise für das Testen der Software
- identifiziert die Produktfunktionen, die von den Tests abgedeckt werden müssen
- beschreibt die Testfälle und Testabläufe, die benötigt werden, um Tests zu bestehen
- spezifiziert die Kriterien für das Bestehen oder das Verfehlen der einzelnen Produktfunktionen.

2.3 Testfallspezifikation

Testfallspezifikation dokumentiert

- die zu benutzenden Eingabewerte und erwarteten Ausgabewerte
- die zu durchführenden Benutzeraktionen und die Reaktion der Anwendung.

2.4 Testablaufspezifikation

Beschreibt alle Schritte zur Durchführung der spezifizierten Testfälle und Implementierung des zugehörigen Test-Designs.

2.5 Testdurchführung

Während einer Testausführung wird für Gewöhnlich ein Testprotokoll geführt, in dem die Testereignisse aufgezeichnet werden.

Alternativ kann man sich auf einen Testabweichungsbericht (Fehlerprotokoll) beschränken. Ein Fehlerprotokoll beschreibt alle Ereignisse, die während einer Testausführung aufgetreten sind und weitere Aktivitäten erfordern.

2.6 Testauswertung

Nach dem Beenden der Testphase wird ein **Testabschlussbericht** verfasst. Er fasst alle Testaktivitäten und derer Ergebnisse zusammen, die mit einer oder mehreren Testentwurfsspezifikationen zusammenhängen.

Der **Testobjektübergabebericht** beschreibt die Übergabe der Testfälle für den Fall, dass getrennte Entwicklungs- und Testteams eingebunden sind oder für den Fall, dass ein offizieller Zeitpunkt für den Beginn einer Testausführung erwünscht ist.