Inhalt

[1. Zweck dieses Dokuments 2](#_Toc16066855)

[2. Umfang 2](#_Toc16066856)

[3. Softwareüberblick 3](#_Toc16066857)

[3.1. Comos 3](#_Toc16066858)

[3.2. Weitere 3rd Party Software 3](#_Toc16066859)

[3.3. Behandlung von Quelldateien / Dokumenten 3](#_Toc16066860)

[3.4. Zielobjekte 3](#_Toc16066861)

[3.5. Oberfläche 3](#_Toc16066862)

[4. Testmerkmale 4](#_Toc16066863)

[4.1. Zu testende Merkmale 4](#_Toc16066864)

[4.1.1. 3rd Party Software Integration 4](#_Toc16066865)

[4.1.2. Allgemein: Fehlerhafte Eingaben 4](#_Toc16066866)

[4.1.3. Zugriffe aufs Dateisystem 4](#_Toc16066867)

[4.1.4. Oberfläche 4](#_Toc16066868)

[4.2. Zu erfüllende funktionale Anforderungen 4](#_Toc16066869)

[4.3. Nicht zu testende Merkmale 4](#_Toc16066870)

[4.3.1. 3rd-Party Software 4](#_Toc16066871)

[4.3.2. Get / Set Methoden 4](#_Toc16066872)

[5. Testabnahme & -kriterien 5](#_Toc16066873)

[5.1. Quellcode-Abdeckung 5](#_Toc16066874)

[5.2. Testergebnisse 5](#_Toc16066875)

[6. Vorgehensweise 6](#_Toc16066876)

[6.1. Integrationstests 6](#_Toc16066877)

[6.2. Komponententests 6](#_Toc16066878)

[6.3. Funktionstests 6](#_Toc16066879)

[7. Testfälle 6](#_Toc16066880)

# Zweck dieses Dokuments

Das vorliegende Dokument stellt den Testplan des 2D-Codereaders dar und unterliegt ständiger Anpassung. Es beinhaltet Definitionen für einen organisierten und systematischen Softwaretest. Es werden zu testende Komponenten und Anwendungsfälle benannt, deren Funktionalität inhärent sicherzustellen ist. Folglich gilt es Testkriterien und Rahmenbedingungen für erfolgreiche Tests zu formulieren.

Der Testplan richtete sich gleichermaßen an Kunden, Entwickler und Tester.

# Umfang

Schwerpunkte der Testfälle sollen Funktionalitäts- und Integrationstests sein, die nach Möglichkeit durch automatisierte Unit-Testfälle abzudecken sind. In diesen Fällen sind deren Spezifikationen direkt ersichtlich. Als Grundlage und Abnahmetests dienen die in der Leistungsspezifikation und Handbuch beschriebenen Anwendungsfälle.

# Softwareüberblick

Dieser Abschnitt stellt einen architekturellen Überblick über den 2D-Codereader dar und beschreibt Zusammenhänge und Abhängigkeiten der einzelnen Komponenten / Module.

## Comos

Die primäre Anwendung des 2D-Codereaders erfolgt als Plugin innerhalb Comos. Insofern ist Comos als Rahmen für Integrationstests zu betrachten. Die größten Berührungspunkte sind das Hosting der Anwendung durch Comos und die Nutzung der Comos Businesslogik mittels COM-Schnittstelle.

Betroffene Komponenten sind von der eigentlichen Kernanwendung im Comos Helper gekapselt. Einzige Ausnahme ist das Comos-Hauptformular.

## Weitere 3rd Party Software

Der 2D-Codereader setzt die Bibliothek QS der Firma Qualitysoft ein. Diese wird mittels ISourceFile, CodeData etc. gekapselt.

## Behandlung von Quelldateien / Dokumenten

## Zielobjekte

## Oberfläche

# Testmerkmale

Der folgende Abschnitt definiert sowohl den Umfang der zu testenden Merkmale des 2D-Codereaders, als auch explizite Ausschlüsse. Zudem erfolgt eine Auflistung welche Merkmalskombinationen relevant sind.

## Zu testende Merkmale

### 3rd Party Software Integration

Der 2D-Codereader setzt die QS-Soft Library als Komponente zum Lesen und Bearbeiten von Matrix-Barcodes ein. Diese kann sich naturgemäß unberechenbar verhalten. Aus diesem Grund ist eine Schnittstelle klar differenziert mit Vorgaben zu testen. Der Umgang mit verschiedenen PDF-Versionen ist ebenfalls Teil eines Tests.

### Allgemein: Fehlerhafte Eingaben

### Zugriffe aufs Dateisystem

### Oberfläche

## Zu erfüllende funktionale Anforderungen

## Nicht zu testende Merkmale

### 3rd-Party Software

3rd-Party Software wird nicht explizit getestet. Das hat der jeweilige Hersteller bereits übernommen. Das schließt sie allerdings nicht von notwendigen Integrationstests in den 2D-Codereader aus.

### Get / Set Methoden

Get- & Set-Methoden – in C# Properties -, die ausschließlich auf interne Attribute zugreifen sind nicht zu testen. Ausnahmen bilden hier Properties, die in seltenen Fällen Logik enthalten.

# Testabnahme & -kriterien

Bugs, „Features“ und Fehler sind inhärenter Bestandteil von Software. Aus diesem Grund sind Kriterien des durchzuführenden Testumfangs zu bestimmen, nach deren Erfüllung Tests als abgeschlossen gelten. Kriterien gliedern sich in folgende Typen: Quellcode-Abdeckung und konkrete Testergebnisse.

## Quellcode-Abdeckung

Bei White-Box-Tests – der Quellcode ist als bekannt vorausgesetzt – sind Testfälle zu definieren, die sämtliche Zeilen des Quellcodes abdecken. Das lässt sich mit separaten Tools quantitativ messen. Anzustrebendes Ziel: 90% Abdeckung.

## Testergebnisse

Fehlgeschlagene manuelle oder automatische Tests dürfen keine kritischen oder katastrophalen Fehler enthalten. Das sind Fehler, die definierte Funktionalitäten einschränken, vollständig verhindern oder zu totalem Versagen führen.

# Vorgehensweise

Der Einsatz von automatisierten Tests ist grundsätzlich manuellen Tests vorzuziehen. Während der Entwicklung entstehende Klassen sind als Blackbox-Test zu entwerfen und nach der Implementierung durch Whitebox-Tests zu ersetzen. Diese testgetriebene Entwicklung beinhaltet anschließend implizites Wissen der Entwickler und deckt frühzeitige Fehler auf.

## Integrationstests

Berührungspunkte an fremde Software durch Komponenten lassen sich nach dem Bottom-Up-Prinzip testen.

## Komponententests

Dem Bottom-Up-Prinzip folgend werden Komponenten in Reihenfolge ihrer Abhängigkeiten getestet. Welche Klassen werden zusammen getestet, bevor weitere hinzukommen. Das Vorgehen lässt sich in Form eines Abhängigkeitsbaumes darstellen.

## Funktionstests

Anforderungsspezifikationen sind sofern möglich automatisiert abzudecken. Teilweise können Ergebnisse nicht automatisiert überprüft werden. Beispielsweise sind zusammengefügte Dokumente durch einen Tester zu bewerten.

Grundsätzlich gilt zusätzlich: Anwendungsfälle werden sie über das grafische Interface manuell getestet und das erwartbare Ergebnis überprüft.

# Testfälle

Komponenten-, Integrations-, Funktions- und Leistungstests werden in der Liste für Testfälle aufgeführt.