



Björn Frauen

(bfrauen@stagil.de)

beschäftigt sich mit modernen, werkzeugunterstützten Lösungsansätzen für Software-, IT- und Business Teams. Er ist Geschäftsführer und Atlassian Expert bei STAGIL und begleitet Unternehmen aus allen Branchen bei der Transformation von Projekt-, Dokumenten- und Testmanagement.

Agiles Testen für Teams mit praxisnahen Ansätzen

Testen ist wie Fitness: Wir wissen, dass moderne Geräte gepaart mit den richtigen Konzepten uns helfen, erfolgreich zu sein – und doch scheitert es oft an der Umsetzung im Alltag. Der Standard des Softwaretestens erhöhte sich zwar in den letzten Jahren. Doch dynamische Einflussfaktoren stören die neue Stabilität: Wellen der „Agilisierung“ brandeten an den idyllischen Strand vieler Softwareteams und schäumten das ruhige Wasser auf.

Immer mehr Unternehmen setzen bei der Erstellung hochwertiger Softwarelösungen auf standardisierte Werkzeuge. Diese sind für viele Teams unverzichtbar, um die produktionsorientierte Bereitstellung umfangreicher Softwarelösungen zu organisieren. Neben der systematischen Verwendung von Prinzipien und Methoden stehen gerade diese Softwareprodukte selbst für professionelle Rahmenbedingungen im Softwareproduktionsprozess. Dabei offenbart besonders ein Kernprozess noch Potenzial für effektivere und effizientere Software- und Systementwicklung: der Softwaretest.

Genauere Kommunikation, durchgängige Transparenz und aufbereitete Live-Daten sind übliche Erwartungen an Anforderungs-, Produktions- und Testteams. Diese stellen bestehende Konzepte infrage und suchen nach neuen Werkzeugen – oder nutzen die vorhandenen besser. Eines der beliebtesten Werkzeuge für Softwareteams ist *JIRA Software* von Atlassian, das sich als praxisnahe Alternative zu spezieller Testsoftware erfolgreich verbreitet.

Agile Softwareentwicklungsteams meinen mit „Test Management“ (TM) zumindest

zwei grundlegende Elemente: Einmal den im Implementierungsprozess integrierten Entwicklertest und darüber hinaus die strukturierte, wiederholbare Verwendung von vordefinierten Tests als Teil eines „Test Case Management“ (TCM).

Agiles Testen

Agiles Testen verfolgt nun den Ansatz, die Anforderung, die Entwicklung und das Testen enger zu verbinden und eine Durchgängigkeit herzustellen. Dieses soll aus Gründen der Transparenz, Dokumentierbarkeit und



Abb. 1: Einordnung von möglichen Integrationen für JIRA Software nach Typ sortiert. (Quelle: Eigene Darstellung von Produkten und Einträgen im Atlassian Marketplace)

| Kriterien | JIRA Software | Erweiterbarkeit |
|--|---------------|-----------------|
| Einfache Anwendbarkeit | ✓ | |
| Unterstützung agiler Methoden | ✓ | |
| Flexibilität und Konfigurierbarkeit | ✓ | |
| Funktionen für Import und Export/API | ✓ | ✓ |
| Wiederverwendbarkeit von Testplänen | (✓) | ✓ |
| Testschritte und Fortschrittsanzeige | (✓) | ✓ |
| Testabdeckung von Anforderungen | (✓) | ✓ |
| Integration in andere Werkzeuge | (✓) | ✓ |
| Unterstützung automatisierter Tests | | ✓ |
| Unterstützung explorativer Methoden | | ✓ |
| Aufnahme und Dokumentation von Testsitzungen | | ✓ |

Table 1: Basisanforderungen an ein Werkzeug für Test Case Management.

zertifizierbaren Nachvollziehbarkeit mit einer modernen Anwendung abgebildet werden, um lokale oder dezentrale Dokumentation in Excel- und Word-Dateien oder E-Mails zu vermeiden.

Eine gute Idee, doch die Probleme der praktischen Umsetzung werden schnell deutlich: Viele Teams nutzen einen eher traditionellen Softwareentwicklungs- und Projektmanagementprozess. Dieser sieht vor, dass Entwickler neue Funktionen produzieren und diese für den Test an das zuständige Testteam übergeben. Das Testteam beginnt nun, die Änderungen zur geplanten Softwareversion zu testen und erstellt anschließend Fehlermeldungen für das Softwareteam. Dieser Prozess klappt für einfache Produkte gut. Ist das zu testende Produkt komplexer und besteht es aus verschiedenen Modulen, die in einem System zusammenwirken, gestaltet sich der Test meist überproportional aufwendig.

Dieses führt unweigerlich zu zwei Entscheidungsoptionen: Die neuen Softwarebestandteile werden ohne die geplanten Tests

in die veröffentlichte Version übernommen oder der Termin für die Fertigstellung muss verschoben werden. Da man Zeitverschiebungen sehr direkt wahrnimmt, aber Qualitätsprobleme erst irgendwann später spürbar werden, entscheiden sich nicht wenige Teams für die dann naheliegende Option und verkürzen den Testumfang. Im besten Fall handeln sich die Produkte damit ein schleichendes Qualitätsproblem ein; oft entstehen allerdings gravierende Fehlerkosten und Imageschäden.

Agiles Testen versucht nun, beiden Ebenen eine valide Umsetzungsoption anzubieten. Entwickler nutzen eigene Tests, um grundlegende Funktionen der angepassten Software zu bestätigen und versteckte oder schwer vorhersehbare Funktionsweisen zu überprüfen. Während Softwareentwickler sehr gut Code analysieren und verbessern können, sind die Kenntnisse bestimmter Anforderungsschnitte, zum Beispiel wie der Anwender bestimmte Funktionen erwartet, nur teilweise bekannt. Daher werden Tester benötigt, die weniger die einzelnen

Codeänderungen im Detail kennen, aber die fachliche Funktionsweise der Anwendung strukturiert überprüfen können. Beide Rollen benötigen ein Werkzeug, das die beteiligten Prozesse integriert und eine einfache Handhabung des Test Management ermöglicht.

Werkzeuge

JIRA Software von Atlassian erfüllt als Standardwerkzeug agiler Softwareteams genau diesen Bedarf. Der klassische Anwendungsbereich kann für agiles Testen erweitert werden. Neben den bekannten Integrationen für Code und Dokumentation verstärken Erweiterungen den Funktionsumfang entweder innerhalb der Standardanwendung selbst oder in einer integrierten Speziallösung (siehe Abbildung 1).

Softwareteams stellen dabei die in Tabelle 1 zusammengefassten typischen Basisanforderungen an ein Werkzeug für Test Case Management, die eine Nutzung für Anforderer, Entwickler und Tester gleichermaßen ermöglichen soll.

Über die wichtigsten Basisanforderungen hinaus werden regelmäßig folgende Detailanforderungen gestellt:

- Strukturen:** Die Abbildung von navigationsunterstützten Strukturen ist, aus einer traditionellen Gewohnheit der Ordnung heraus, eine beliebte Funktion zur Objektverwaltung. Als Gegenstück zu einer flachen Objektstruktur, wie diese in vielen filterbasierten Werkzeugen vorgehalten wird, bietet eine visuelle, ordnerähnliche Struktur viele Vorteile für die tägliche Arbeit (siehe Abbildung 2). Häufig werden Produktmodule als Gliederung für Tests

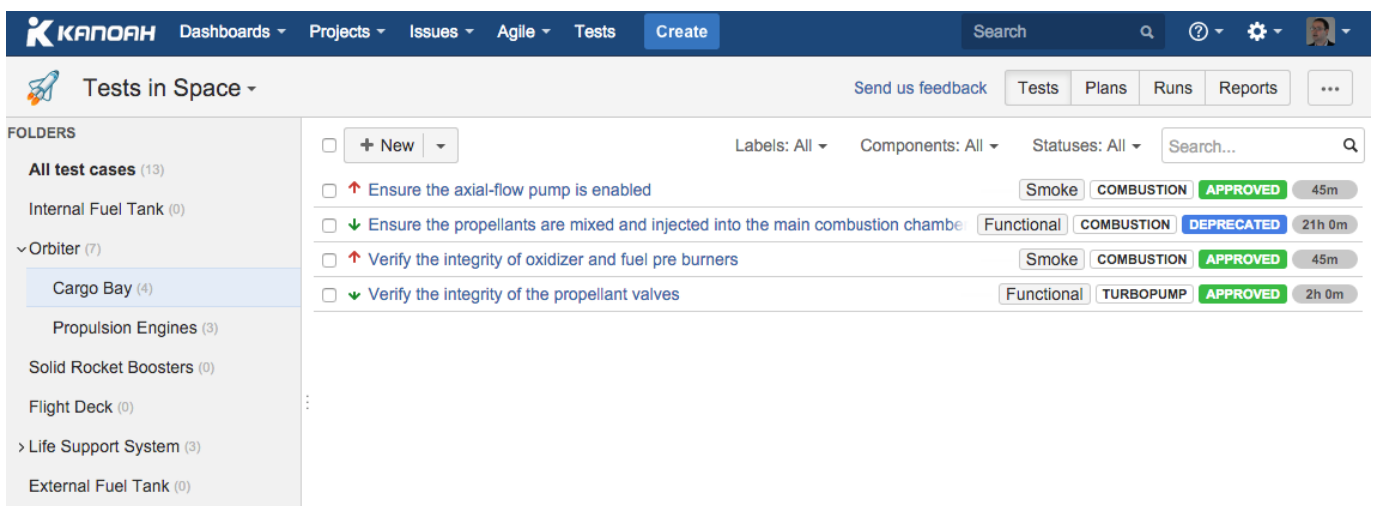


Abb. 2: Ansicht einer Struktur in JIRA Software mit der Erweiterung Kanoah Tests. (Quelle: Atlassian Marketplace)

The screenshot shows the qTest interface for a test run titled "Verify Historical Record of Reports". The interface includes a sidebar with a project tree, a top navigation bar with tabs like "TEST PLAN", "REQUIREMENTS", "TEST DESIGN", "TEST EXECUTION", "SESSIONS", "QTEST INSIGHTS", and "REPORTS". The main area displays the test run details, including a table of execution history and a detailed test log.

| NO. | TESTER | DATA SOURCE | NAME | VERSION | TARGET RELEASE/BUILD | EXECUTION TYPE | CONFIGURATION | EXECUTED | STATUS |
|-----|-------------|-------------|-------------------------------------|---------|----------------------|----------------|---------------|------------------------|---------|
| 11 | Ryan Yackel | | Verify Historical Record of Reports | 6.0 | Release 1 / Build 1 | Functional | Chrome + OS X | 05/04/2016 12:09:08 PM | FAILED |
| 10 | Ryan Yackel | | Verify Historical Record of Reports | 6.0 | Release 1 / Build 1 | Functional | Chrome + OS X | 05/04/2016 12:08:59 PM | BLOCKED |
| 9 | Ryan Yackel | | Verify Historical Record of Reports | 6.0 | Release 1 / Build 1 | Functional | Chrome + OS X | 05/04/2016 12:08:49 PM | PASSED |
| 8 | Ryan Yackel | | Verify Historical Record of Reports | 6.0 | Release 1 / Build 1 | Functional | Chrome + OS X | 05/04/2016 12:08:43 PM | PASSED |
| 7 | Ryan Yackel | | Verify Historical Record of Reports | 6.0 | Release 1 / Build 1 | Functional | Chrome + OS X | 05/04/2016 12:07:25 PM | FAILED |
| 6 | Ryan Yackel | | Verify Historical Record of Reports | 4.0 | Release 1 | Functional | Chrome + OS X | 10/28/2015 1:59:54 PM | FAILED |
| 5 | Ryan Yackel | | Verify Historical Record of Reports | 4.0 | Release 1 | Functional | Chrome + OS X | 10/28/2015 1:43:02 PM | FAILED |
| 4 | Ryan Yackel | | Verify Historical Record of Reports | 4.0 | Release 1 | Functional | Chrome + OS X | 10/28/2015 1:41:58 PM | FAILED |
| 3 | Ryan Yackel | | Verify Historical Record of Reports | 3.0 | Release 1 | Functional | Chrome + OS X | 10/22/2015 7:41:48 AM | FAILED |
| 2 | Ryan Yackel | | Verify Historical Record of Reports | 3.0 | Release 4 | Functional | Chrome + OS X | 10/19/2015 1:24:10 PM | PASSED |

| STEP # | DESCRIPTION | EXPECTED RESULT | ACTUAL RESULT | STATUS |
|--------|--|---------------------------------|---------------|--------|
| 1 | (Home Login Session) [Nordstrom Online & In Store: Shoes, Jewelry, Clothing, Makeup, Dresses] Click 'Men' link | Home Page Appears | | PASSED |
| 2 | (Home Login Session) Click 'Polos' link | Polo Shirts for Men Nordstrom | | PASSED |
| 3 | (Home Login Session) Select 'Black' combo item | | | PASSED |
| 4 | (Login Script - Called Test Case) Called Step 1 | | Actual Result | FAILED |
| 5 | (Home Login Session) [Nordstrom Online & In Store: Shoes, Jewelry, Clothing, Makeup, Dresses] Click 'Men' link | Home Page Appears | | PASSED |
| 6 | (Home Login Session) Click 'Polos' link | Polo Shirts for Men Nordstrom | | PASSED |
| 7 | (Home Login Session) Select 'Black' combo item | | | PASSED |
| 8 | (Login Script - Called Test Case) Called Step 1 | | Actual Result | FAILED |

The screenshot shows a JIRA bug report for "Sample Scrum Project B / RDY-106". The bug title is "Log into the application under test and make sure that you can see the historical comments from 1 week prior to today's date." The status is "READY FOR QA". The priority is "High (P2)". The description includes steps to reproduce the issue: 1) Log in to Nordstrom, 2) Click 'Polos' link, 3) Select 'Black' combo item, 4) Call Step 1. The test execution table shows that steps 1, 2, 3, and 5-7 passed, while steps 4 and 8 failed.

Details

- Type: Bug
- Priority: High (P2)
- Affects Version/s: None
- Labels: None
- Environment: QA - 001
- Status: READY FOR QA (View Workflow)
- Resolution: Unresolved
- Fix Versions: None

Description

- [Nordstrom Online & In Store: Shoes, Jewelry, Clothing, Makeup, Dresses] Click 'Men' link
Expected Result: Home Page Appears
- Click 'Polos' link
Expected Result: Polo Shirts for Men | Nordstrom
- Select 'Black' combo item
- Called Step 1

Actual Result: Actual Result

qTest: Test Execution

Project: Demo - JIRA QASProduct

| Test Log # | Executed Start | Executed End | Status |
|------------|----------------------|----------------------|---------|
| 11 | 04/May/2016 13:09 PM | 04/May/2016 13:09 PM | FAILED |
| 10 | 04/May/2016 13:08 PM | 04/May/2016 13:08 PM | BLOCKED |
| 9 | 04/May/2016 13:08 PM | 04/May/2016 13:08 PM | PASSED |
| 8 | 04/May/2016 13:08 PM | 04/May/2016 13:08 PM | PASSED |
| 7 | 04/May/2016 13:07 PM | 04/May/2016 13:07 PM | FAILED |

Abb. 3: Ansicht einer Planung integriert mit JIRA Software in der Erweiterung qTest. (Quelle: Atlassian Marketplace)

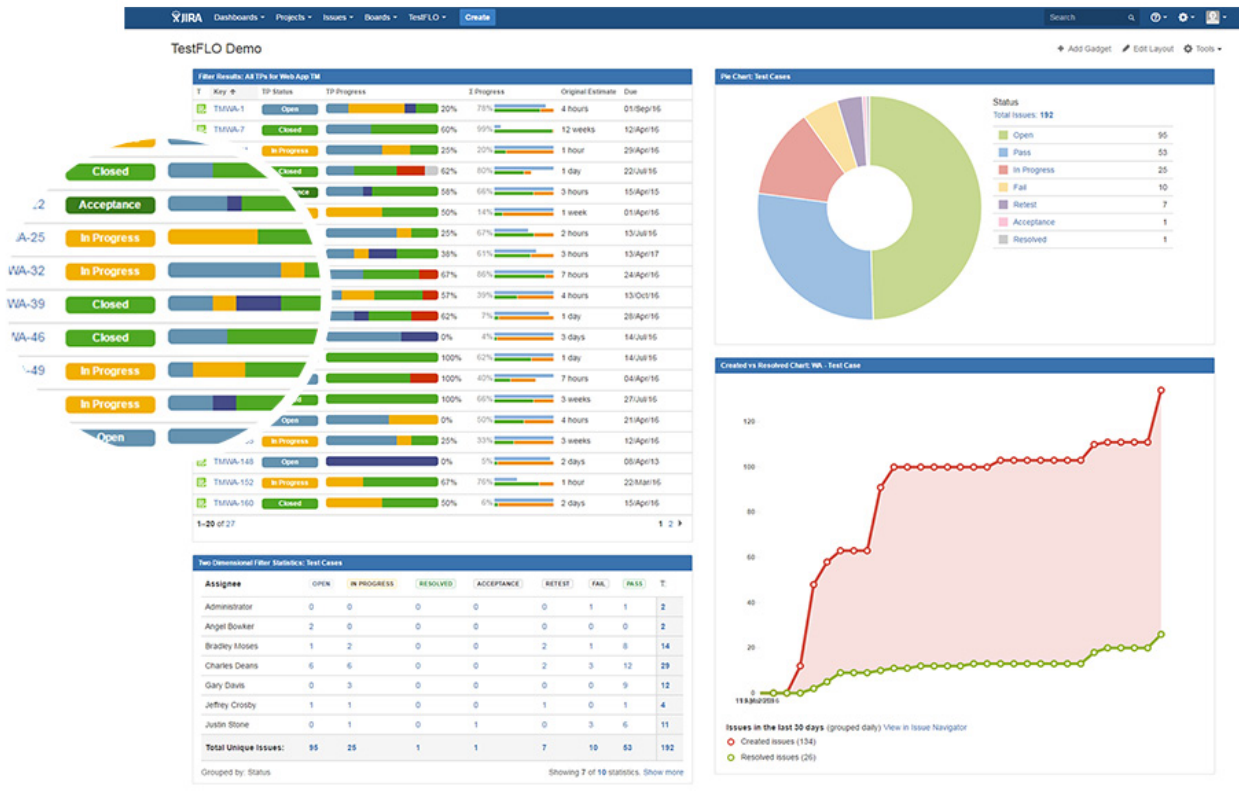


Abb. 4: Ansicht eines Live-Reports in JIRA Software mit der Erweiterung TestFLO. (Quelle: Atlassian Marketplace)

genutzt. Einzelne Testpläne sind so leichter zuzuordnen und lassen sich später einfacher auffinden. Alternativ werden Testphasen als Struktureinheit verwendet.

- **Planung:** Am Anfang jedes Testvorhabens steht die Auswahl und Bereitstellung geeigneter Testfälle. Eine Planungssicht (siehe Abbildung 3) muss daher wichtige Informationen kompakt auf einen Blick darstellen und den Arbeitsfluss im Testteam begleiten.
- **Berichte:** Viele Testteams wechseln von Excel oder manuell gestalteten Berichten in ein Werkzeug, das Live-Daten aus dem Produktionsprozess nutzt (siehe Abbildung 4). Neben der Auflistung der Anforderungen, Funktionen oder Tests sollen einzelne Attribute, wie Status oder Zeiterfassung, konsolidiert werden. Der Verlauf von Metriken für Testpläne oder Defekte kann so aus den Echtzeit-Informationen visualisiert werden. Diese Übersichten dienen als Basis für erweiterte, kommentierte Berichte für Präsentationen und Projekttreffen.

Erfahrungen

Projekte zur Modernisierung des Test Management können mit unterschiedlichen

Ergebnissen als erfolgreich bewertet werden. Für ein Team mit sehr einfachen Werkzeugen und Methoden liefert die reine Einführung eines modernen Werkzeugs für Test Case Management und einer angeschlossenen Agilisierung der Prozesse einen großen Fortschritt. Andere Teams arbeiten vielleicht schon sehr lange mit einem professionellem Test Management und fokussieren sich daher auf feingranulare Potenziale.

Die einfache Erstellung und Wiederverwendbarkeit von Testplänen kann mit wenig Aufwand in JIRA Software abgebildet werden. Je nach eingesetzter Erweiterung und der individuellen Handhabung stehen Teststrukturen in wenigen Minuten bereit. Die Planung und die Umsetzung neuer Testdurchläufe können ohne unnötige Verzögerungen dokumentiert starten. Der Aufwand für die Verwaltung und die Wartezeit für das Testteam kann so regelmäßig halbiert werden.

Viele Teams schätzen zudem die durch JIRA Software bereitgestellten Abläufe und vordefinierten Testschritte. Ein Vorteil eines teamfähigen Werkzeugs besteht in der Nutzung von Echtzeit-Informationen, wie einer konsolidierten Fortschrittsanzeige aller Aktivitäten. So stehen viele Auswertungen, die bisher manuell erstellt oder aufwendig nachbereitet wurden, jederzeit und kurzfristig zur Verfügung. Damit sparen Testteams wertvolle Zeit.

Zusammenfassung

Die Erwartungen an agiles Testen sind erheblich und lassen sich wie folgt aufgliedern:

- **Releases:** Durch häufigere Veröffentlichungen erreichen die nützlichen Produktanpassungen den Kunden schneller und sichern Wettbewerbsvorteile.
- **Flexibilität:** Durch Schaffung günstiger Rahmenbedingungen für Zusammenarbeit und Austausch erzielen die beteiligten Teams bessere Leistungen.
- **Wert:** Durch Erfüllen der steigenden Anforderungen an Zuverlässigkeit und Kosteneffizienz in Zeiten der Digitalisierung.

Um eine hohe Softwarequalität zu sichern, sind sowohl manuelle, explorative als auch automatisierte Tests notwendig. Durch ein TCM-Werkzeug unterstützte Tests erhöhen die Testabdeckung und damit die Codequalität. Die Kosten der späteren Fehlerbehebung sinken zudem deutlich.

Die Auswahl einer passenden TCM-Lösung bedeutet für viele Teams den beschleunigten Einsatz agiler Methoden und zugleich eine modernen Abbildung von Anforderungs- und Testprozessen. Neben dem Funktionsumfang selbst betrifft der gewählte Ansatz eine grundlegende Entscheidung: Soll ein TCM-Werkzeug sehr nah an

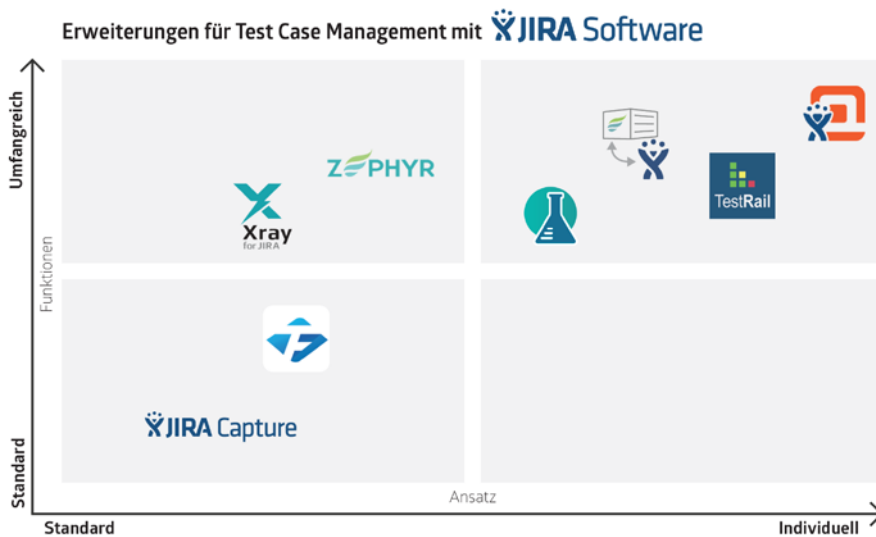


Abb. 5: Einordnung von Erweiterungen für JIRA Software nach Ansatz und Funktionsumfang. (Quelle: Eigene Darstellung von Einträgen im Atlassian Marketplace)

der eingesetzten Standardlösung etabliert werden oder ist eine speziellere Lösung notwendig. Beispielsweise ist es sinnvoll, die Notwendigkeit einer Unterstützung explorativer Methoden oder weitreichender Automatisierung zu diskutieren (siehe [Abbildung 5](#)).

Egal für welche Lösung sich Teams schlussendlich entscheiden, agiles Testen kann mit praxisnahen Ansätzen umgesetzt werden. Eine Veränderung, wie im Fall der eigenen Fitness, ist nur selten einfach. Doch lohnt sich diese oder ist sogar notwendig, um langfristig erfolgreich zu sein. ■