



Testbericht

erstellt von

Daniel Böhm

Abgabe

09.08.2009

Auftraggeber

Rent-A-Jet

Prüfer

Prof. Dr. Dr. H. Neunteufel

Hochschule Wismar

Fachbereich Wirtschaft

Inhaltsverzeichnis

1.	Vorwort.....	3
2.	Vorgehensweise.....	4
2.1.	Planung.....	4
2.2.	Durchführung.....	5
2.3.	Auswertung.....	6
2.3.1.	Testprotokoll.....	7
3.	Fazit.....	9

1 Vorwort

“Panta rei – alles fließt” sagt Heraklit. Die Zeiten ändern sich und sind in einem schnelllebigen Wandel unterzogen. Im Hinblick auf die Softwareentwicklung ist diese Aussage von besonderer Bedeutung. Nicht nur die Anforderungen des Kunden, auch die technischen Möglichkeiten, die Benutzeroberfläche der Programme und die Kommunikation zwischen Mensch und Maschine sind in einem ständigen Wandel unterzogen. Alles fließt nur nicht im gleichen Takt, sondern schneller. Auch das Testen von Softwareapplikationen bleibt von diesem Wandel nicht unberührt. Denn das Testen von Software zielt nicht darauf ab, alle Funktionalitäten zu testen, sondern die wichtigsten. Daher wurden Methoden und Testszenarien entwickelt, um die relevantesten Fälle zu analysieren und Fehler zu finden.

Aufgrund des zeitlich begrenzten Projektrahmens galt es die schwerwiegendsten Fehler schnell zu finden. James Bach, der Autor mehrerer Bücher zum Thema Testen, beschreibt die unterschiedlichen Vorgehensweisen beim sogenannten Rapid Testing. Das Rapid Testing unterscheidet sich vom herkömmlichen Testen auf folgende Weise.

Zentrale Aufgabe des Rapid Testen ist die Fehler möglichst schnell zu finden und ist auch primärer Ausgangspunkt des Testen. Nicht die Erstellung von Testszenarien steht also hier im Mittelpunkt. Weiterhin erfordert Rapid Testing die Fertigkeiten und das Wissen zu Testzwecken optimal einzusetzen, um aus fehlerhaften Szenarien logische Fakten ziehen zu können. Der Fokus beim Rapid Testing zielt auf schwerwiegende und wichtigsten Probleme innerhalb der Softwareapplikation ab.

Auf der Homepage von James Bach www.satisfice.com betont er die Notwendigkeit den Testgegenstand während der Testingphase zu erforschen bzw. zu erlernen. Diese Feststellung wird durch seine Aussage “Rapid testing is also rapid learning” weiterhin untermauert. Der ursprünglich ausgeführte Test bildet die Grundlage für alle darauffolgenden Testszenarien.

Dies wurde ebenfalls bei der Durchführung der Rent-A-Jet AG Software berücksichtigt. Z.B. wurde bei Anlegen eines neuen Kunden nachfolgend das entsprechende Angebot angelegt, um festzustellen ob sämtliche Daten des Kunden bei der Angeboterstellung berücksichtigt werden.

Im Allgemeinen kann gesagt werden, dass Tester und Entwickler in einer wechselseitigen Relation zueinander stehen. Die Tester versuchen auf der einen Seite Fehler zu finden, und die Entwickler auf der anderen Seite sind dannach bestrebt, das Erfolgserlebnis der Tester möglichst gering zu halten.

Dieser vorliegende Testbericht dokumentiert die Vorgehensweise des Testing und dokumentiert im Anhang die relevantesten Fehlerszenarien in tabellarischer Form.

2 Vorgehensweise

Während des Testings bzw. der Planungsphase des Testings wurde die “drei-Stufen-Theorie” verwendet, welches den Testablauf in verständlicher Weise interpretiert. Diese sind:

- Planung – Worum geht es eigentlich?
- Durchführung – Wie wurde getestet?
- Auswertung – Was wäre das korrekte Ergebnis?

2.1 Planung

Im Rahmen der Planungsphase wurde festgelegt was getestet werden sollte. Als Ausgangsposition dienen die Vorgaben von der Rent-A-Jet AG. Nachstehend werden die Vorgaben der Rent-A-Jet AG zusammengefasst:

Workflow Charterflug:

- 1) Anfrage seitens des Kundenanforderungen
- 2) Finanzielles Angebot erstellen und verschicken
- 3) Bei Vorliegen der Antwort
 1. Wenn Antwort positiv, Vertrag vorbereiten
 2. Wenn Antwort negativ, Gründe erfragen → dann Ende
- 4) Wenn Vertrag unterschrieben zurück
 1. Flugzeug, Crew und Catering bereitstellen
 2. Rechnung erstellen und verschicken, Zahlung verfolgen
 3. Flug durchführen
 4. Kundenzufriedenheit erfragen und abspeichern

Workflow: Kostenverfolgung

- 1) Zahlungseingang verfolgen
- 2) Wenn nicht zum Termin
 1. Mahnwesen

Analysen:

- 1) Kundenzufriedenheit analysieren
- 2) Ablehnungsgründe der Angebote analysieren
- 3) Profitabilität der Flugzeuge analysieren

Weiterhin galt es folgende Funktionalitäten in der Testing-Planung zu berücksichtigen, die die Berechnung des Angebots betreffen:

- Anzahl der nötigen Zwischenlandungen (pro Landung: Erhöhung der Charterzeit um 45 Min)
- Kosten (Anteil an Fixkosten plus Anteil an Personalkosten plus hourly cost*Flugzeit)

Weitere Testing Szenarien ergaben sich in der zusätzlichen Softwarefunktionalität:

- Ausgabe des Angebotes als Brief auf Word oder PDF oder direkt als e-mail an den Interessenten (mit Bild des Flugzeuges)
- Ausgabe des Vertrages (ohne Bild)
- Ausgabe der Rechnung

Im Anbetracht der Tatsache, dass das Projekt zeitlich eingeschränkt ist, stellte sich für ThinkLogics die Frage, wie die Testprioritäten festgelegt werden sollten. Neben dem methodischen Vorgehen von Testprioritäten ist auch das intuitive Vorgehen ein wichtiger Bestandteil des Testens. Da ThinkLogics ausschließlich mit erfahrenen Testern arbeitet, sind diese in der Lage die entsprechenden Schwachstellen bzw. mögliche potentielle Fehlerquellen aufzudecken.

Im ersten Schritt wurde dabei auf ein besonderes Augenmerk auf den allgemeinen Workflow gesetzt. D.h. es sollte sichergestellt werden, dass z.B. bei der Anfrage eines Kunden bis zur Vertragserstellung sämtliche Daten eingegeben werden können und diese auch in der Angebotserstellung reflektiert werden. Hauptfunktionen wie die Anfrage eines Kunden, Vertragsvorbereitung sowie Rechnungserstellung waren Hauptfaktoren von Testinszenarien.

Im Rahmen der allgemeinen Workflowausführung entstanden keine schwerwiegenden Fehler, wie die Testdokumentation im Anhang beweist. Problemfälle entstanden eher im Umfeld der Detailplanung bzw. Ausführung, also die Festlegung von Entfernungen bzw. Berechnungen. Diese Fehler aber bereits behoben, dass diese innerhalb der aktuellsten Softwareversion nicht auftreten sollten.

2.2 Durchführung

Die Durchführung von Testszenarien sind drei Schritte notwendig:

- Testumgebung bereitstellen
- Test ausführen
- Testergebnis ermitteln

Im Rahmen des Regressionstest werden bekannte bereits beseitigte fehlerhafte Testzenarien durchgeführt. Dies ist insbesondere dann der Fall, wenn z.B. die neueste Softwareversion aufgrund eines fehlerhaften Defects installiert werden musste. Die Testumgebung spielt in diesem Zusammenhang eine signifikante Rolle, wenn es um die Wiederholbarkeit von Test geht. ThinkLogics nutze die Testumgebung von Berlios. Dies ermöglichte nachzuvollziehen, welche Person den Fehler entdeckt hat und mit unter welchen Vorbedingungen, mit welchen Aktionen und in welcher Programmversion der Fehler gefunden wurde.

Einen automatisierten Testablauf hat ThinkLogics nicht für notwendig erachtet, da hier ein System mit wenigen Komponenten bereitgestellt wurde.

Im Rahmen der Testausführung wurde dabei ein besonders Augenmerk auf mögliche potentielle bzw. aus der Vergangenheit bekannte Fehlerquellen gesetzt. Dies gewährt die Zuverlässigkeit und Robustheit des Programms daher reichten in diesem Falle einfache Testszenarien bzw. Abläufe aus. Die Testergebnisse wurden in Listenform protokolliert. Bereits wiederholte Fehler wurden im Vorfeld eliminiert.

Im Zusammenhang wird die Durchführung bzw. der Ablauf bei Entdeckung eines fehlerhaften Testzenarios erläutert. In der allgemeinen Fachsprache wird hier der Begriff "Problem Tracking and Reporting" verwendet (kurz PTAR). Ein sogenannter PTAR kann ein technisches oder ein fachliches Problem darstellen. Ein technisches Problem ist immer dann zu bejahen, wenn Teilkomponenten des Systems nicht einwandfrei laufen. Ein fachliches Problem hingegen, stellt einen falschen oder unvollständigen Workflow bzw. unrichtige Berechnungen. Insbesondere im Rahmen der Flugstreckenentfernung wurde das Testing Team von ThinkLogics mit fehlerhaften Berechnungen konfrontiert, die mittlerweile vom Entwicklungsteam bereinigt wurden. Im Allgemeinen lässt sich jedoch die Aussage treffen, dass die fehlerhaften Szenarien eher in der Anwendungsoberfläche lagen. D.h. so war es z.B. nicht möglich eine neue Kundenadresse anzulegen (Defect 015893). Generell gab es bei Vorliegen eines Problem Tracking and Reporting folgende Vorgehensweisen bzw. Ansätze:

- Auffinden eines Defects bzw. Problems
- Erkennen eines Defects
- Soll/ Ist Beschreibung des vorliegenden Defects
- Einbeziehung der Entwicklung
- Kontrolle bzw. Retest nach Korrektur durch Entwicklung
- Ablage bzw. Defect auf Closed setzen

Grundsätzlich lässt sich abschließend sagen, dass die Durchführung von Testszenarien anhand der beschriebenen Workflows in Kapitel XX durchgeführt wurde. Die Test wurden im Test Tracking System Berlios protokolliert und für den Kunden im Kapitel Testprotokoll sämtliche Defects noch einmal in Excel Format gesammelt.

2.3 Auswertung

Die Auswertung eines Testablaufs bedeutet, dass das geforderte fehlerfreie Ergebnis mit dem tatsächlichen Ergebnis verglichen wird. Hierzu dienen die dokumentierten Vorgaben, die bereits in Kapitel XX beschrieben sind. Diese Vorgaben bildeten die Quelle des Testens und somit auch die Auswertung einzelner Testszenarien. Wie bereits geschildert ist im Anhang ein Testprotokoll hinterlegt, welches die einzelnen fehlerhaften Testszenarien protokolliert. Die gesammelten Materialien d.h. Protokolle dienen für die spätere Verwendung oder Nachvollziehbarkeit z.B. in Projekten.

Weiterhin lässt sich bestätigen, dass gerade in kleineren bis mittelgroßen Projekten Testprotokolle bzw. -berichte bedeutend sind für Lernzwecke. Denn Daten über die Entwicklung des Produkts z.B. Fehler in bestimmter Entwicklungsphase werden gesammelt und daraus können entsprechende Rückschlüsse auf die Effizienz des Testprozesses geschlossen werden. Dies kann als eine Grundlage für eine mögliche Verbesserung dienen.

In diesem Zusammenhang ist die Gewichtung einzelner Testszenarien zu unterscheiden. Beginnend mit der Kategorisierung lassen sich Defekts in diverse Kategorien einordnen.

Folgende Defectkategorien waren im Projektverlauf bedeutend:

- Backend
- GUI – Graphical User Interface

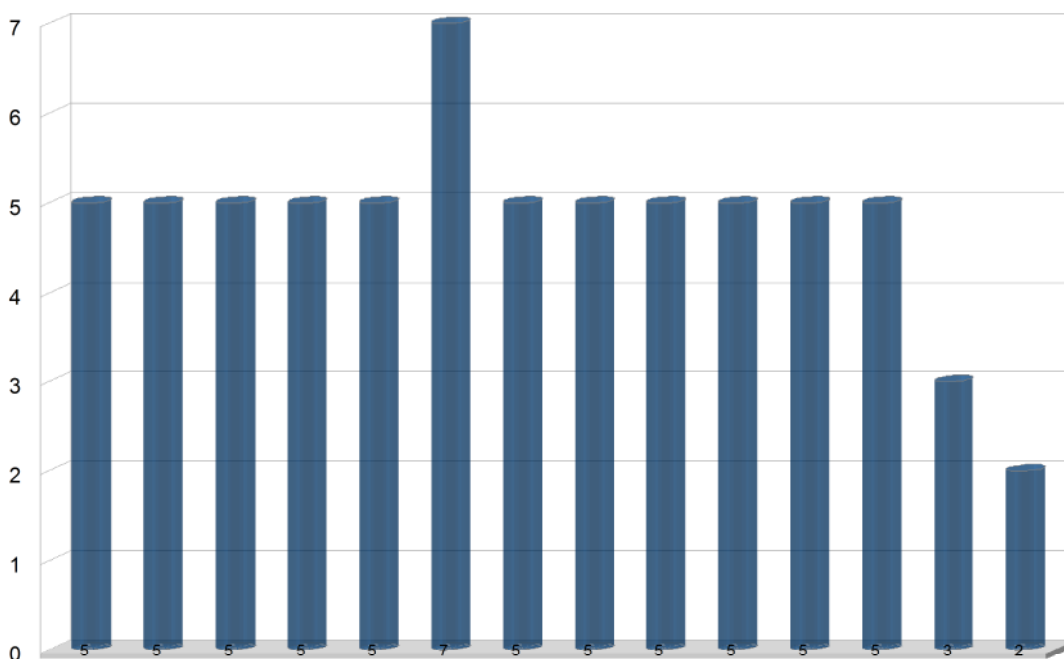
2.3.1 Testprotokoll

Im vorliegenden Testbericht werden sämtliche fehlerhafte Testszenarien bzw. Testabläufe nachstehend erläutert:

Bug ID	Fehlerbeschreibung	Resultat (Fehler)	Kategorie	Priorität	Submitted	Gefixt
15923	Flugzeughersteller anlegen nicht möglich	Fehlermeldung nach anlegen erscheint	Backend	5	24.06.09	25.06.09
15920	MwSt. anlegen teilw. Möglich trotz Fehlermeldung	Fehlermeldung erscheint nach Anlegen des MwSt. Workaround durch Löschen bereits bestehender MwSt dann neu Anlegen.	Backend	5	23.06.09	23.06.09
15919	Zufriedenheitstypen anlegen nicht möglich	Fehlermeldung erscheint	Backend	5	23.06.09	23.06.09
15906	Flugzeugtypen – Änderung Sitze nicht möglich	Fehlermeldung erscheint	Backend	5	22.06.09	22.06.09
15905	Neues Flugzeug anlegen nicht möglich	Fehlermeldung erscheint "SQL Error"	Backend	5	22.06.09	22.06.09
15876	Verträge werden nicht in Pdf Format angezeigt	Meldung „Adobe kann Vertrag nicht öffnen, da der Dateityp nicht unterstützt wird oder die Datei beschädigt ist“	Backend	7	17.06.09	18.06.09
15874	Anlegen einer neuen Adresse nicht möglich	Meldung „Undefined Offset: 0 (CORE/cake/libs/model.php, line 2410)“	Backend	5	17.06.09	18.06.09
15870	Flugzeughersteller ändern – Anpassung des Layouts		Backend	5	16.06.09	16.06.09
15869	Anzahl der Flugbegleiter keine Relevanz für Endbetrag	Preis bleibt bei Anzahl der Flugbegleiter gleich	Backend	5	16.06.09	16.06.09
15863	Website aus Report erstellen	Duplikat zu 15868	Backend	5	15.06.09	16.06.09
15861	Vorgangstypen Informationen auf der Startseite	Keine Informationsanzeige	Backend	5	15.06.09	16.06.09
15860	MwSt Informationen auf der Startseite	Keine Informationen anzeige mit Mauszeiger	GUI	5	15.06.09	17.06.09
15859	Flugplätze Informationen auf der Startseite	Keine Informationsanzeige	GUI	5	15.06.09	18.06.09
15855	Anlegen neuer Mehrwertsteuersätze	Eingabefelder nicht erkennbar	GUI	3	15.06.09	16.06.09
15853	Frontend passt nicht	Frontend nicht angepasst	GUI	2	15.06.09	18.06.09

Wie aus dem Fehlerprotokoll zu entnehmen ist, sind sämtliche Defects mit Prioritätsmerkmalen versehen. Je niedriger die Prioritätseinstufung, desto unbedeutender der Defekt, je höher desto bedeutender der Defekt. Generell wurden die Defects in relativ kürzester Zeit – ggf. noch am selben Tag bearbeitet.

Die nachfolgende graphische Darstellung demonstriert die Anzahl der Defekts nach deren Gewichtung bzw. Priorität.



Während der gesamten Testphase gab es insgesamt 15 zu bearbeitende Defects. Davon ver-einnahmten fehlerhafte Testszenarien mit Priorität 5 den höchsten Anteil – insgesamt wurden diese mit 83,3 % bewertet. Schwerwiegende Fehler – mit Priorität 7 – bildeten hingegen die Ausnahme. Lediglich 9,72 % trugen dazu bei. Defects mit geringerer Priorität verbuchten lediglich 4,17 % (Priorität 3) und 2,78 % (Priorität 2).

3 Fazit

“Testing is a desctructive process, even a sadistic process” (Myers)

Testing ist in der Theorie sehr mathematisch veranlagt. In der Wirtschaftspraxis ist es jedoch nicht annähernd so zielgerichtet, wie es scheint. Das Risiko ist für jede Projektsituation gesondert zu kalkulieren und zu Beginn schwer abzuschätzen. Der enge Zeitrahmen und das eingeschränkte Budget konfrontierte ThinkLogics mit neuen Herausforderungen. Grundsätzlich gilt die Aussage “Je früher ein Fehler entdeckt wird, desto kostengünstiger kann er behoben werden”. Aufgrund dieser Feststellung hat ThinkLogics bereits während der Entwicklungsphase bestimmte Komponenten frühzeitig getestet und entsprechend dokumentiert. Das gewonnene Know-how wird somit im Hinblick auf künftige Testprojekte erhalten bleiben. Relevant in diesem Zusammenhang ist, dass Fehler nicht wiederholt werden sollten, sondern auf dem bereits Erreichten dazulernt und aufbaut.

Wie bereits das vorliegende Testprotokoll darstellt, sind keine schwerwiegende Defekts beispielsweise in Form von erheblichen Systembeeinträchtigungen oder -fehlern vorgekommen. Der Kunde Rent-A-Jet AG erhält eine individualisierte und vor allem wettbewerbsfähige Software, die bereits vor der Einführungs- bzw. Go-Live phase bereitgestellt werden kann.

ThinkLogics ist der einzige Dienstleister für mittelständische Unternehmen, der eine solche Bandbreite an erforderlichen Dienstleistungen liefern kann, um diesen multidisziplinären Herausforderungen gerecht zu werden. Wir haben nicht nur die Erfahrung in der Erstellung und Ausführung von Testprojekten, sondern auch in der Entwicklung sowie des aktiven Projektmanagements. Wir verstehen die IT-Perspektive und haben eine nachweisbare Erfolgsgeschichte. Sprechen Sie uns an – wenn Sie weitere Dienste benötigen.