

Explore It!



Elisabeth Hendrickson (@testobsessed) arbeitet als Testerin, Entwicklerin und »Agile-Enabler«. Im Jahr 1980 schrieb sie ihre erste Codezeile und fand sofort ihre ersten Fehler. 2010 gewann sie den renommierten Gordon Pask Award von der Agile Alliance. Sie ist bekannt für ihren Google Tech Talk über Agile Testing sowie ihre beliebten Testheuristiken-Spickzettel. Sie teilt ihre Zeit auf zwischen lehren, vortragen, schreiben, programmieren und der Arbeit in agilen Teams, die ihr Engagement beim Testen sehr schätzen.



Übersetzerin: Meike Mertsch arbeitet als begeisterte Testerin für Magine AB in Stockholm, Schweden. Sie hat einen Hintergrund als agiler Entwickler und Coach und ein Faible für leichtgewichtige Methoden wie Kanban und Personal Kanban sowie agiles Entwickeln und Testen. In ihrer Freizeit läuft und skatet sie durch Stockholm oder macht die Kletterhallen und Felsen in der Nähe unsicher.

Elisabeth Hendrickson

Explore It!

**Wie Softwareentwickler und Tester
mit explorativem Testen Risiken reduzieren
und Fehler aufdecken**

Aus dem Amerikanischen übersetzt von Meike Mertsch



dpunkt.verlag

Elisabeth Hendrickson
elisabeth@testobsessed.com

Übersetzung: Meike Mertsch, Stockholm
Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Satz: Petra Strauch, just in print, Bonn
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:
Buch 978-3-86490-093-8

1. Auflage, Translation Copyright für die deutschsprachige Ausgabe © 2014 dpunkt.verlag GmbH
Wiebinger Weg 17
69123 Heidelberg

Copyright der amerikanischen Originalausgabe © 2013 The Pragmatic Programmers, LLC.
Title of American original: Explore It! Reduce Risk and Increase Confidence with Exploratory Testing
Pragmatic Bookshelf, The Pragmatic Programmers, LLC, Dallas, Texas, Raleigh, North Carolina
<http://pragprog.com>
ISBN-13: 978-1-937785-02-4

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

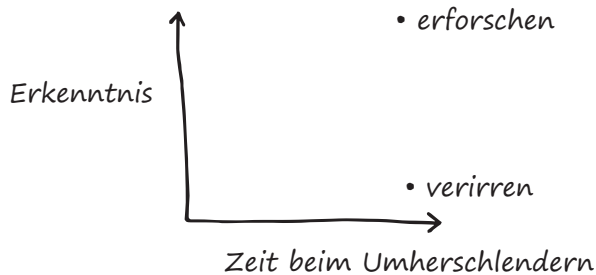
Vorwort

Um irgendetwas wirklich zu verstehen, müssen Sie es erforschen¹.

Das gilt zum Beispiel für Städte. Wenn ich reise, dann plane ich immer ein wenig Zeit ein, um in den kleinen Gassen umherzuschlendern und versteckte Kostbarkeiten zu finden. Ich meide Touristenbereiche, um die Restaurants zu finden, in die nur die Einheimischen gehen, und die Geschäfte, die alltägliche Dinge führen. Auf diesen verschlungenen Pfaden lerne ich die örtliche Kultur kennen und mache mich mit dem wahren Charakter des Ortes vertraut.

Das gilt auch für Software. Wenn Sie mehr über die wahre Leistungsfähigkeit und Pfade Ihrer Software lernen möchten, müssen Sie sich abseits der ausgetretenen Wege bewegen.

Es gibt aber einen Unterschied zwischen ziellosem Umherschlendern und wahren Erforschen, wie Jessica Hagy so gewandt in ihrem Webcomic »Indexed« mit dem Titel »Field Notes« (dt.: Feldnotizen, Aufzeichnungen) festhielt².



Jessicas Comic kommuniziert eine wichtige Warnung: Wenn Sie ohne Richtung oder Zweck umherlaufen, werden Sie viel Zeit mit Schlendern für herzlich wenig Erkenntnisse verbringen: Sie werden sich verirren.

1. »Forschen« (engl. to explore) klingt im Deutschen in Bezug auf Software zunächst etwas fremd. Da exploratives Testen von Software sich aber von anderen Testarten unterscheidet, bei denen gezielt die Software geprüft und untersucht wird, kann man hier durchaus davon sprechen, in der Software zu forschen bzw. die Software zu erforschen.
2. Comic von Jessica Hagy, mit freundlicher Genehmigung (<http://thisisindexed.com/2011/10/field-notes/>)

Dieses Buch handelt davon, wie man erfolgreich forscht. Auf diesen Seiten werden sie die grundlegenden Fähigkeiten und Techniken eines guten Forschers entdecken.

Für wen ist dieses Buch?

Da dieses Buch vom Softwaretesten handelt, könnten Sie denken, dass es für Tester geschrieben wurde. Ist es auch, aber nicht nur. Ich habe dieses Buch für jeden geschrieben, der daran interessiert ist, verlässliche und robuste Software zu schreiben. Das schließt Programmierer, Business-Analysten, Produktmanager und sogar solche Leute mit ein, die Software normalerweise erst dann zu Gesicht bekommen, wenn sie fast fertig ist: den Support.

Wenn Sie Tester mit fundiertem Wissen über Testdesigntechniken wie Zustandsanalysen sind, werden Sie lernen, wie Sie diese Fähigkeiten anwenden können, um Tests während des Erkundens der Software »on-the-fly« zu entwerfen.

Wenn Sie Programmierer mit einem tiefen Verständnis für die grundlegende Technologie sind, werden Sie die Software aus verschiedenen Blickwinkeln betrachten und lernen, sie aus unterschiedlichen Perspektiven zu untersuchen.

Wenn Sie Business-Analyst oder Produktmanager sind, werden Sie lernen, wie Sie Ihr Zusammenspiel mit der Software variieren, um sicherzustellen, dass die Software auch unter unterschiedlichsten Bedingungen das tut, was Sie beabsichtigt haben.

Wenn Sie zum Support gehören, dann werden Sie erkennen, wie Sie Ihr Verständnis der Software vertiefen, für die Sie Hilfe anbieten. Sie werden auch ein paar Tricks kennenlernen, wie man schwer reproduzierbare Kundenprobleme nachstellen kann.

Kurzum, wenn Sie im weiteren Sinne zur Softwareentwicklung gehören, spezifizieren, programmieren, verifizieren oder Support anbieten, ist dieses Buch für Sie.

Es macht auch keinen Unterschied, an welcher Art von Software Sie arbeiten. Die Techniken in diesem Buch lassen sich auf einem weiten Gebiet technischer Umgebungen von Webapplikationen, Desktop-Anwendungen, mobilen Applikationen über Embedded-Echtzeitsoftware bis hin zu APIs und Webservices anwenden.

Wie dieses Buch aufgebaut ist

Dieses Buch besteht aus drei Teilen:

- Teil 1, *Grundlagen schaffen*, führt grundlegende Bausteine für Fähigkeiten im Untersuchen ein. In diesem Abschnitt werden Sie lernen, Testcharter zu erstellen, um Ihre Untersuchungen zu leiten, wie Sie beobachten können, was wirklich passiert (Hinweis: Das ist schwieriger, als es klingt), wie Sie interessante

Variationen erkennen und wie Sie feststellen, welches Verhalten zu erwarten ist, wenn Software auf eine Art und Weise benutzt wird, über die vorher noch niemand nachgedacht hat.

- Teil 2, *Weitere Möglichkeiten ins Spiel bringen*, baut auf diesen Grundlagen auf. Sie werden lernen, wie Sie Software untersuchen, indem Sie Interaktionen, Sequenzen, Daten, Zeitabläufe und Konfigurationen ändern. Auf dem Weg werden Sie erfahren, wozu Analysetechniken wie Zustandsmodelle, Datenmodelle und Kontextdiagramme beim Untersuchen nützlich sein können.
- Teil 3, *Zusammenhänge herstellen*, überträgt diese Techniken in das Umfeld eines Softwareprojekts. Sie werden lernen, wie Sie die Ideen aus Teil 1 und 2 in verschiedenen Situationen nutzen können, beispielsweise um bereits bestehende (Legacy-)Applikationen oder Software zu erforschen, die kein User Interface besitzt. Sie werden erfahren, wie Sie Ihre Funde mitteilen können und wie Sie die Untersuchung schon ganz am Anfang des Entwicklungszyklus einsetzen können.

Obwohl Sie die Kapitel in jeder beliebigen Reihenfolge lesen können, ziehen Sie den größten Erkenntnisgewinn aus jedem Teil, wenn Sie die Konzepte in den jeweils vorangehenden Teilen verinnerlicht haben.

Praxis ist unerlässlich für den Aufbau Ihrer Fähigkeiten im explorativen Testen. Um Ihnen dabei zu helfen, endet jedes Kapitel mit einem Praxisabschnitt. Damit können Sie das Wissen um die Konzepte festigen und Ideen generieren, wie Sie die Konzepte des Kapitels direkt anwenden können.

Im gesamten Buch werden Sie spezielle Techniken oder Heuristiken finden. Jede Heuristik hat einen Namen, wie »Einige, keine, alle« oder »Anfang, Mitte, Ende«. Namen von Heuristiken werden in Anführungszeichen geschrieben und Sie finden die komplette Liste in Anhang 2, *Testheuristiken-Spickzettel*.

Onlinequellen

Wie alle Titel von Pragmatic Bookshelf hat dieses Buch eine Webseite. Wenn Sie <http://pragprog.com/book/ehxta/explore-it> besuchen, können Sie an Diskussionen mit anderen Lesern und mir teilnehmen. Außerdem können Sie Fehler melden (wenn Sie welche finden, hoffe ich sehr, dass Sie sich die Zeit nehmen und mir davon berichten).

Viel Spaß beim Forschen!

Elisabeth Hendrickson
elisabeth@testobsessed.com
Februar 2013