

Testbarkeitsfaktoren und Testaufwand: Auswertung dreier Umfragen

Stefan Jungmayr

Version 1.1
28. April 2006

Abstract

Um den Testaufwand unter Kontrolle zu halten und die Testeffektivität zu erhöhen ist es wichtig, die Testbarkeit der Software gezielt zu verbessern. Eine Reihe von Verbesserungsmaßnahmen wurde bisher beschrieben, es fehlen bislang aber Grundlagen für eine gezielte Priorisierung dieser Maßnahmen. Dieser Artikel beschreibt den Aufbau sowie die Ergebnisse dreier Umfragen zur Wirkung von Testbarkeitsfaktoren auf den Testaufwand. Untersucht wurden Testbarkeitsfaktoren mit Bezug zu den Anforderungen, zum Programm und zu Prozessaspekte mit Auswirkung auf die Anforderungsdokumente. Wesentliches Ergebnis der Umfrage ist, dass den Anforderungen die größte Wirkung auf den Testaufwand zugesprochen wurde und der Beobachtbarkeit bzw. Kontrollierbarkeit eine größere Wirkung als der Programm-Komplexität.

Inhalt

1	Einleitung	3
2	Testbarkeit der Anforderungen.....	3
3	Testbarkeit des Programms	5
4	Prozess-Unterstützung.....	7
5	Vergleich der Gruppen von Testbarkeitsfaktoren.....	8
6	Zusammenfassung und Diskussion.....	8
Anhang A	Fragebogen	10
Anhang B	Daten	11
Anhang C	Testbarkeitsfaktoren in alternativer Sortierung.....	12
Anhang D	Referenzen	13

Änderungshistorie

Version	Datum	Kommentar
1.0	28.02.2006	Initiale Version
1.1	28.04.2006	Einarbeitung der Ergebnisse der Umfrage beim ASQF-Treffen vom 28.3.2006. Änderung des Titels gemäß der Anzahl der Umfragen. Fehleinträge in Tabelle 3 korrigiert.

1 Einleitung

Der Testaufwand umfasst etwa 25% bis 50% des gesamten Entwicklungsaufwands [Spil03]. Um den Testaufwand zu reduzieren bzw. die Testeffektivität zu erhöhen ist es wichtig, die Testbarkeit der Software gezielt zu verbessern. Testbarkeit ist der Grad, zu dem ein Software-Produkt den Test in einem gegebenen Testkontext ermöglicht. Einige Ansätze zur Verbesserung der Testbarkeit wie z.B. die testgetriebene Entwicklung [Link05] und die anforderungsgetriebene Testbarkeitsentwicklung [Jung05] wurden bereits beschrieben. Es fehlen aber Informationen über den Einfluss der Testbarkeit auf den Testaufwand, sodass für eine Priorisierung der Verbesserungsmaßnahmen keine solide Grundlage vorhanden ist

Die in diesem Artikel beschriebenen Umfragen haben das Ziel, den Einfluss unterschiedlicher Faktoren auf den Testaufwand zu bewerten. Die Umfragen mit insgesamt 51 Teilnehmern sind inhaltlich größtenteils identisch und wurden zu drei Zeitpunkten durchgeführt:

- **Umfrage A** fand am 17. November 2005 in München im Rahmen des 23. Treffens der Fachgruppe TAV [Url:TAV] der Gesellschaft für Informatik [Url:GI] mit 24 Umfrageteilnehmern (vorwiegend Tester, Testmanager und Hochschullehrer) statt.
- **Umfrage B** fand am 29. und 30. November 2005 an der Technischen Akademie Esslingen (TAE) im Rahmen des Kolloquiums „Testen im Entwicklungsprozess“ mit 16 Umfrageteilnehmern statt. Der Fragebogen war aufgrund des Feedbacks von Umfrage A leicht adaptiert und erweitert worden.
- **Umfrage C** fand am 28. März 2006 im Rahmen des Treffens der Fachgruppe „Software-Test“ des ASQF mit 11 Umfrageteilnehmern statt und ist inhaltlich mit Umfrage B identisch

Die Unterschiede zwischen den Umfragen A und B bzw. C sind aus Tabelle 2 in Anhang A ersichtlich, welche alle Fragen und Erläuterungen aus den Fragebogen enthält.

Die Teilnehmer konnten bei jeder Frage die Wirkung auf den Testaufwand wie folgt bewerten:

- hoch,
- mittel,
- niedrig,
- keine Wirkung.

Untersucht wurden in den Umfragen schwerpunktmäßig Testbarkeitsfaktoren aus den folgenden Bereichen:

- Testbarkeit der Anforderungen,
- Testbarkeit des Programms,
- Prozessaspekte mit Auswirkung auf die Anforderungsdokumente.

Die folgenden Kapitel 2 bis 4 enthalten die kombinierte Auswertung der Antworten für die Umfragen A und B. Die Einzelergebnisse für Umfrage A, B und C sind aus Tabelle 3 in Anhang B ersichtlich.

2 Testbarkeit der Anforderungen

Die in Dokumenten festgehaltenen Anforderungen sind die Grundlage für die Testfallspezifikation. Dementsprechend haben Eigenschaften der Anforderungen einen starken Einfluss auf den Aufwand von Testaktivitäten wie z.B. Testfalldefinition und Testauswertung.

Folgende Testbarkeitsfaktoren beziehen sich auf die Testbarkeit der Anforderungen und wurden bei der Umfrage berücksichtigt:

- Komplexität der Anforderungen,
- Abhängigkeiten zwischen Anforderungen,
- testbare Formulierung der Anforderungen,
- Priorisierung der Anforderungen.

Abbildung 1 enthält die Beschreibung der Testbarkeitsfaktoren, wie sie in der Umfrage gegeben wurde.

<p>1.1 Komplexität Die Bestimmung des erwarteten Testergebnisses ist schwierig oder zu aufwendig (z.B. bei Simulation, NP-hartes Problem, komplexe graphische Operation).</p> <p>1.2 Abhängigkeiten Anforderungen können nicht isoliert geprüft werden, da sie zu eng verwoben sind.</p> <p>1.3 Testbare Formulierung Anforderungen können nicht objektiv geprüft werden, da sie nicht quantitativ formuliert bzw. unzureichend detailliert sind.</p> <p>1.4 Priorisierung Die Anforderungen sind nicht priorisiert. Eine Priorisierung der Testfälle ist daher nicht möglich.</p>

Abbildung 1: Beschreibung der Testbarkeitsfaktoren bzgl. Anforderungen

Ergebnisse: Die Umfrageteilnehmer gaben an, dass die Komplexität der Anforderungen sowie die Abhängigkeiten zwischen den Anforderungen den größten Einfluss auf den Testaufwand haben (Abbildung 2). Der testbaren Formulierung von Anforderungen wurde eine geringere Bedeutung zugemessen, der Priorisierung der Anforderungen die geringste Wirkung.

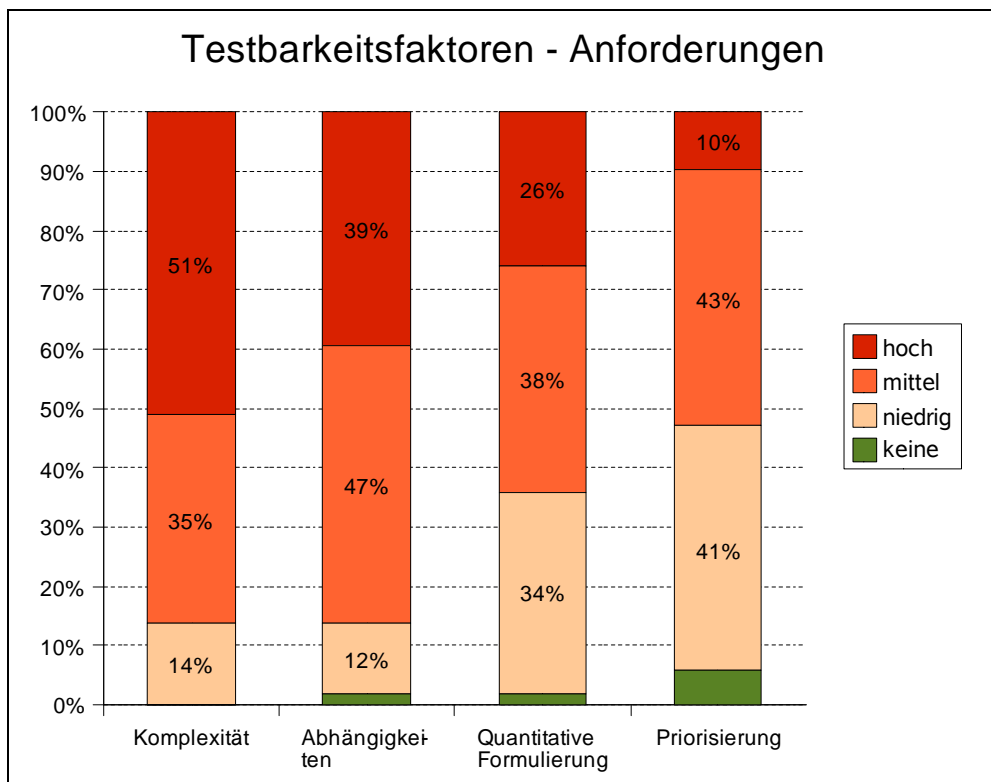


Abbildung 2: Antworten zu Testbarkeitsfaktoren bzgl. Anforderungen

3 Testbarkeit des Programms

Folgende Testbarkeitsfaktoren beziehen sich auf die Testbarkeit des Programms und wurden bei der Umfrage berücksichtigt:

- Komplexität,
- Isolierbarkeit,
- Trennung der Verantwortlichkeiten,
- Kontrollierbarkeit,
- Beobachtbarkeit,
- Verständlichkeit,
- Automatisierbarkeit,
- Diagnostizierbarkeit,
- Fehler-Lokalisierbarkeit,
- Robustheit,
- Heterogenität.

Abbildung 3 enthält die Beschreibung der Testbarkeitsfaktoren, wie sie in der Umfrage gegeben wurde.

2.1 Komplexität

Die Komplexität der Realisierung (z.B. Komplexität von Schnittstellen, Kontrollfluss, Vererbung, Verteiltheit, Parallelität) ist im Verhältnis zum Anwendungsproblem zu hoch, die praktisch erreichbare Überdeckung zu gering.

2.2 Isolierbarkeit

Das Testobjekt kann aufgrund von Abhängigkeiten nicht isoliert getestet werden.

2.3 Trennung der Verantwortlichkeiten

Die Testobjekte haben keine klar abgrenzte Verantwortlichkeit.

2.4 Kontrollierbarkeit

Für ein Testobjekt bzw. eine Testressource (z.B. Datenbank) kann z.B. folgendes nicht erreicht werden: 1) für den Test erforderlicher Initialzustand, 2) gewünschter Kontrollfluss, 3) gewünschter Ausgabewert, 4) deterministisches Systemverhalten.

2.5 Beobachtbarkeit

Die Test(zwischen)ergebnisse können nicht oder nur schwer beobachtet werden, bzw. werden (z.B. bei Echtzeitsystemen) durch die Beobachtung selbst verfälscht und damit ungültig.

2.6 Verständlichkeit

Die Test(zwischen)ergebnisse können nicht (ausreichend) interpretiert werden (z.B. durch impliziten Kontrollfluss, fehlende Dokumentation, Performanz-Tuning).

2.7 Automatisierbarkeit

Tests können nicht ausreichend automatisiert werden oder dauern zu lange. Die Automatisierung wird nicht ausreichend durch verfügbare Testtools unterstützt.

2.8 Diagnostizierbarkeit

Das System kann seine Integrität nicht durch einen Selbst-Test selbst prüfen. Der Tester kann an das System keine selbst-definierten Abfragen bzgl. des Systemzustands stellen.

2.9 Fehler-Lokalisierbarkeit

Fehler lassen sich nicht ausreichend der verursachenden Software-Einheit zuordnen.

2.10 Robustheit

Nach dem Auftreten von Systemfehlern können die verbleibenden Tests nicht oder nur mehr eingeschränkt ausgeführt werden.

2.11 Heterogenität

Eine Vielzahl von Hardware-, Middleware-, Software- und Entwicklungs-Plattformen, Programmiersprachen etc. erhöht den Testaufwand.

Abbildung 3: Beschreibung der Testbarkeitsfaktoren bzgl. des Programms

Ergebnisse: Abbildung 4 zeigt die Testbarkeitsfaktoren, sortiert nach der Antwortkategorie „hoch“. (Anmerkung für den interessierten Leser: Anhang C enthält eine alternative Sortierung der Testbarkeitsfaktoren.)

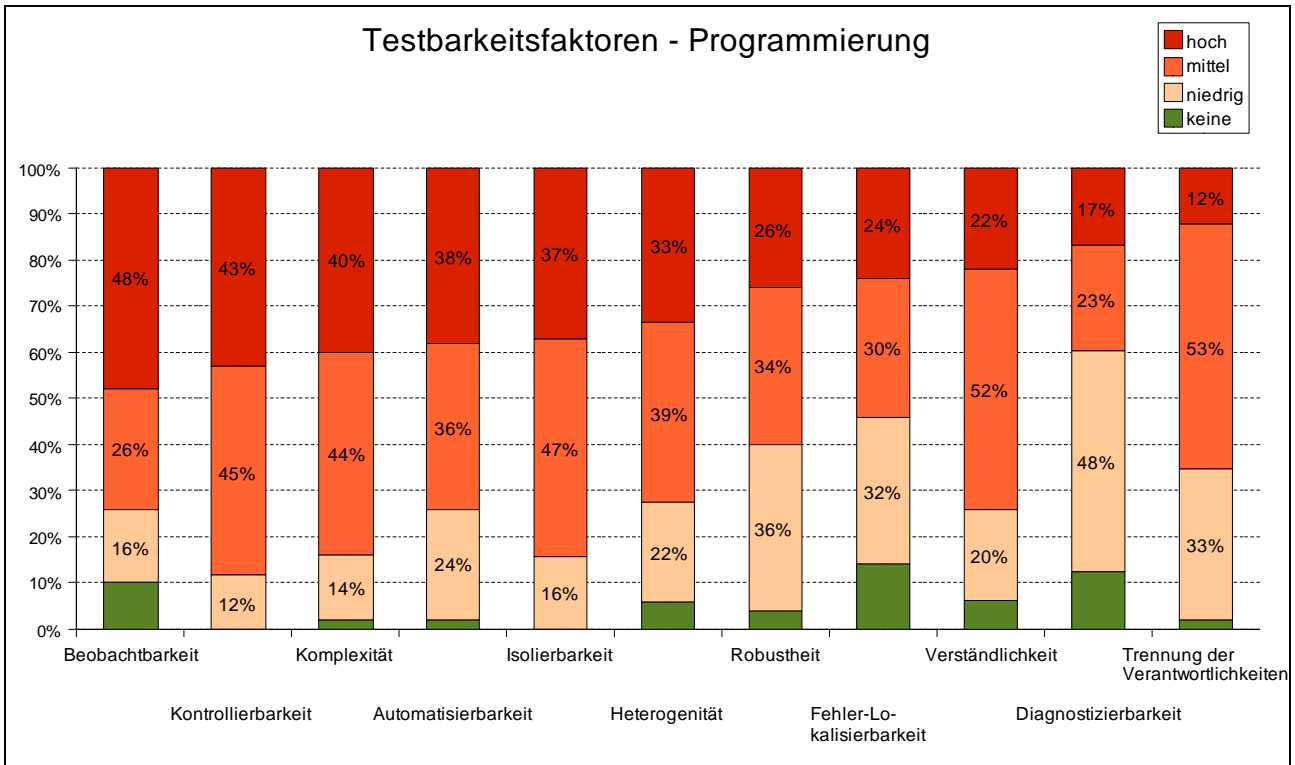


Abbildung 4: Antworten zu Testbarkeitsfaktoren bzgl. der Programmierung

Aufgrund der Teilnehmerantworten bildet der Autor die in Tabelle 1 dargestellten vorläufigen Gruppen von Testbarkeitsfaktoren hinsichtlich ihrer Wirkung auf den Testaufwand

Wirkung auf Testaufwand	Testbarkeitsfaktor bzgl. Programmierung
größte Wirkung	Beobachtbarkeit Kontrollierbarkeit
zweitgrößte Wirkung	Komplexität Automatisierbarkeit Isolierbarkeit Heterogenität
drittgrößte Wirkung	Robustheit Fehler-Lokalisierbarkeit Verständlichkeit
viertgrößte Wirkung	Diagnostizierbarkeit Trennung der Verantwortlichkeiten

Tabelle 1: Rangfolge der Testbarkeitsfaktoren bzgl. Programmierung

Weitere Untersuchungen sind sicherlich notwendig, um u.a. die Rangfolge bei enger zusammenliegenden Faktoren abzusichern oder zu verschieben.

4 Prozess-Unterstützung

Auch der Entwicklungsprozess hat einen Einfluss auf den Testaufwand. Im Rahmen der Umfrage wurden schwerpunktmäßig Faktoren untersucht, die einen direkten Bezug zur Testreferenz haben sowie zwei Faktoren zu Testumfang und Testdaten:

- Verfügbarkeit der Testreferenz (nur Umfrage B und C),
- Verfolgbarkeit der Testreferenz,
- Aktualität der Testreferenz,
- Stabilität der Testreferenz,
- Klarheit des Testumfangs (nur Umfrage B und C),
- Verfügbarkeit der Testdaten (nur Umfrage B und C).

Abbildung 5 enthält die Beschreibung der Testbarkeitsfaktoren, wie sie in der Umfrage gegeben wurde.

<p>3.1 Verfügbarkeit der Testreferenz Die Anforderungen sind nicht dokumentiert.</p> <p>3.2 Verfolgbarkeit der Testreferenz Die Anforderungen und der Entwurf zu einem Softwaremodul sind schwer oder gar nicht aufzufinden.</p> <p>3.3 Aktualität der Testreferenz Die Dokumentation der Anforderungen bzw. der Realisierung ist nicht aktuell.</p> <p>3.4 Stabilität der Testreferenz Die Änderungsrate der Anforderungen bzw. der Realisierung ist zu hoch.</p> <p>3.5 Testumfang Der Umfang der zu testenden Anforderungen im aktuellen Release ist unklar (z.B. bei iterativen Entwicklungsprozessen).</p> <p>3.6 Testdaten Die notwendigen Testdaten sind nicht ausreichend vorhanden bzw. zugänglich.</p>

Abbildung 5: Beschreibung der Testbarkeitsfaktoren bzgl. Prozessunterstützung

Ergebnisse: Abbildung 6 zeigt die Antworten der Teilnehmer, sortiert nach der Anzahl der Antworten „hoch“. Demnach haben die Verfügbarkeit, Verfolgbarkeit und Stabilität der Testreferenz einen größeren Einfluss auf die Testkosten als die die Verfügbarkeit der Testdaten, die Aktualität der Testreferenz und die Klarheit des Testumfangs.

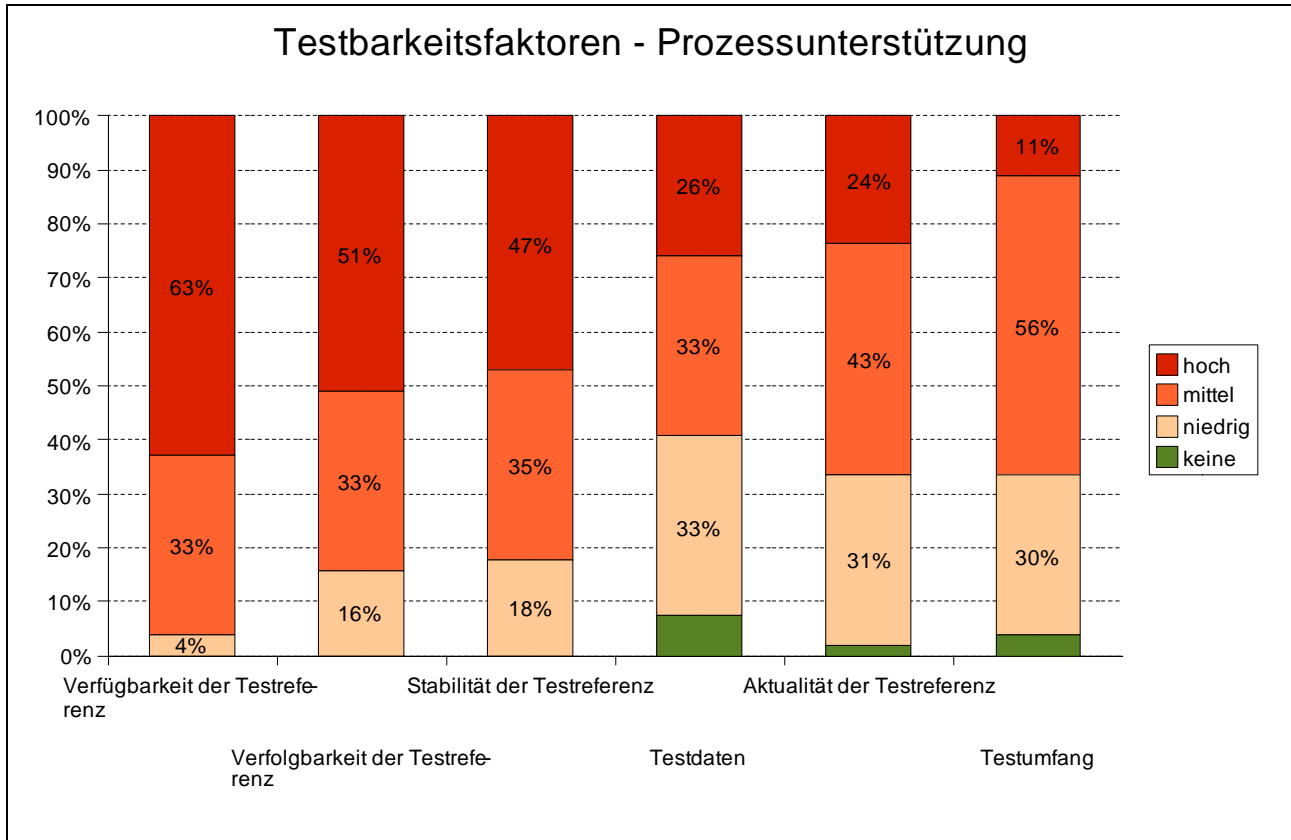


Abbildung 6: Antworten zu Testbarkeitsfaktoren bzgl. Prozessunterstützung

5 Vergleich der Gruppen von Testbarkeitsfaktoren

Vergleicht man die Prozentzahlen der Kategorie „hoch“ der stärksten Testbarkeitsfaktoren zwischen den Gruppen (d.h. zwischen den Testbarkeitsfaktoren bzgl. Anforderungen, Programmierung und Prozessunterstützung), so kann man feststellen:

- Die Testbarkeitsfaktoren zur Prozessunterstützung haben die höchsten Werte bei der Antwortkategorie „hoch“ erzielt: 63%, 51%, etc. (siehe Abbildung 6).
- Die Testbarkeitsfaktoren zu den Anforderungen haben die zweithöchsten Werte bei der Antwortkategorie „hoch“ erreicht: 51%, 39%, etc. (siehe Abbildung 2). Die Testbarkeitsfaktoren zur Programmierung bekamen Bewertungen in ähnlicher Höhe, beginnend mit 48% und 43% (siehe Abbildung 4).

6 Zusammenfassung und Diskussion

Die kleine Stichprobengröße lässt noch keine endgültigen Aussagen zu, die folgenden Trends lassen sich aber feststellen:

Den größten Einfluss auf die Testkosten sprechen die Umfrageteilnehmer den Anforderungen zu, wobei die Verfügbarkeit, Verfolgbarkeit und Stabilität der Anforderungen höher bewertet wird als die eigentlichen Eigenschaften der Anforderungen wie Komplexität und Abhängigkeiten.

Bei den Eigenschaften des Programms bewerteten die Umfrageteilnehmer die Kontrollierbarkeit und Beobachtbarkeit am stärksten mit der Kategorie „hoch“. Erst in zweiter Linie sollen die Kosten von der Heterogenität, Isolierbarkeit, Komplexität und Robustheit der Testobjekte abhängen. Dass

die Komplexität meist nicht an erster Stelle genannt wird, ist ein interessantes Ergebnis dieser Umfrage.

Um die Ergebnisse dieser Studie zu validieren sind weitere Umfragen mit mehr Teilnehmern erforderlich. Für die praktische Anwendung der Ergebnisse wäre auch eine Differenzierung der Statistik z.B. nach der Anwendungsdomäne (eingebettete Systeme, Echtzeitsysteme, Informationssysteme) oder Kritikalität der Software nützlich.

Empirische Untersuchungen zur Wirkung von Testbarkeitsfaktoren auf den Testaufwand wären erstrebenswert – bis aber diese in ausreichender Zahl vorliegen, werden wohl noch einige Jahre vergehen. In der Zwischenzeit können Tester und Entwickler Umfragen wie diese als eine pragmatische Informationsquelle nutzen.

Hinweis

Weitere Informationen zum Thema Testbarkeit sowie Kontaktmöglichkeiten zum Autor bietet folgende Website: <http://www.testbarkeit.de>

Danksagungen

Vielen Dank an Falk Fraikin, Matthias Hamburg und Mario Winter für ihre Reviews und Vorschläge zu Draft-Versionen des Fragebogens und dieses Artikels. Ebenso vielen Dank an die Teilnehmer der Umfrage für ihre Unterstützung und Mitwirkung!

Anhang A Fragebogen

Tabelle 2 enthält die Testbarkeitsfaktoren sowie die Erläuterungen für Umfrage A und Umfrage B.

	Faktor		Erläuterung	
	Umfrage A	Umfrage B, C	Umfrage A	Umfrage B, C
1	Testbarkeit der Anforderungen			
1.1	Komplexität		Die Bestimmung des erwarteten Testergebnisses ist schwierig oder zu aufwendig (z.B. bei Simulation, NP-hartes Problem, komplexe graphische Operation).	
1.2	Abhängigkeiten		Anforderungen können nicht isoliert geprüft werden, da sie zu eng verwoben sind.	
1.3	Quantitative Formulierung	Testbare Formulierung	Anforderungen können nicht objektiv geprüft werden, da sie nicht quantitativ formuliert sind.	Anforderungen können nicht objektiv geprüft werden, da sie nicht quantitativ formuliert bzw. unzureichend detailliert sind.
1.4	Priorisierung		Die Anforderungen sind nicht priorisiert. Eine Priorisierung der Testfälle ist daher nicht möglich.	
2	Testbarkeit des Programms			
2.1	Komplexität		Die Komplexität der Realisierung (z.B. Komplexität von Schnittstellen, Kontrollfluss, Vererbung, Verteiltheit, Parallelität) ist im Verhältnis zum Anwendungsproblem zu hoch, die praktisch erreichbare Überdeckung zu gering.	Die Komplexität der Realisierung (z.B. Komplexität von Schnittstellen, Kontrollfluss, Vererbung, Verteiltheit, Parallelität) ist im Verhältnis zum Anwendungsproblem zu hoch.
2.2	Isolierbarkeit		Das Testobjekt kann (ohne Eingriff in den Source-Code) nicht isoliert getestet werden (z.B. durch zyklische Abhängigkeiten, hartkodierte Abhängigkeiten).	Das Testobjekt kann aufgrund von Abhängigkeiten nicht isoliert getestet werden.
2.3	Trennung der Verantwortlichkeiten		Die Testobjekte haben keine klar abgrenzte Verantwortlichkeit.	
2.4	Kontrollierbarkeit		Für ein Testobjekt bzw. eine Testressource (z.B. Datenbank) kann z.B. folgendes nicht erreicht werden: 1) für den Test erforderlicher Initialzustand, 2) gewünschter Kontrollfluss, 3) gewünschter Ausgabewert, 4) deterministisches Systemverhalten.	
2.5	Beobachtbarkeit		Die Test(zwischen)ergebnisse können nicht oder nur schwer beobachtet werden, bzw. werden (z.B. bei Echtzeitsystemen) durch die Beobachtung selbst verfälscht und damit ungültig.	
2.6	Verständlichkeit		Die Test(zwischen)ergebnisse können nicht (ausreichend) interpretiert werden (z.B. durch impliziten Kontrollfluss, fehlende Dokumentation, Performanz-Tuning).	
2.7	Automatisierbarkeit		Tests können nicht ausreichend automatisiert werden oder dauern zu lange. Die Automatisierung wird nicht ausreichend durch verfügbare Testtools unterstützt.	
2.8	Diagnostizierbarkeit		Das System kann seine Integrität nicht durch einen Selbst-Test selbst prüfen. Der Tester kann keine selbst-definierte Abfrage an das System stellen.	Das System kann seine Integrität nicht durch einen Selbst-Test selbst prüfen. Der Tester kann an das System keine selbst-definierten Abfragen bzgl. des Systemzustands stellen.
2.9	Fehler-Lokalisierbarkeit		Fehler lassen sich nicht ausreichend der verursachenden Software-Einheit zuordnen..	
2.10	Robustheit		Nach dem Auftreten von Systemfehlern können die verbleibenden Tests nicht oder nur mehr eingeschränkt ausgeführt werden.	
2.11	Heterogenität		Eine Vielzahl von Hardware-, Middleware-, Software- und Entwicklungs-Plattformen, Programmiersprachen etc. erhöht den Testaufwand.	
3	Prozess-Unterstützung			
3.1	-	Verfügbarkeit der Testreferenz	-	Die Anforderungen sind nicht dokumentiert.
3.2	Verfolgbarkeit der Testreferenz		Die Anforderungen und der Entwurf zu einem Softwaremodul sind schwer oder gar nicht aufzufinden. Werden Anforderungen geändert, so lassen sich die dadurch notwendigen Änderungen an den Testfällen nur schwer identifizieren.	Die Anforderungen und der Entwurf zu einem bestimmten Softwaremodul sind schwer oder gar nicht zu identifizieren.
3.3	Aktualität der Testreferenz		Die Dokumentation der Anforderungen bzw. der Realisierung ist nicht aktuell.	
3.4	Stabilität der Testreferenz		Die Änderungsrate der Anforderungen bzw. der Realisierung ist zu hoch.	
3.5	-	Testumfang	-	Der Umfang der zu testenden Anforderungen im aktuellen Release ist unklar (z.B. bei iterativen Entwicklungsprozessen).
3.6	-	Testdaten	-	Die notwendigen Testdaten sind nicht ausreichend vorhanden bzw. zugänglich.

Tabelle 2: Testbarkeitsfaktoren mit Erläuterung

Anhang B Daten

Tabelle 3 beschreibt die Anzahl der Antworten pro Testbarkeitsfaktor und Antwortkategorie für die Umfragen A und B sowie für die Kombination der Daten beider Umfragen.

	Faktor		Umfrage A				Umfrage B				Umfrage C				Umfrage A + B + C			
	Umfrage A	Umfrage B + C	keine	nied.	mittel	hoch	keine	nied.	mittel	hoch	keine	nied.	mittel	hoch	keine	nied.	mittel	hoch
1	Testbarkeit der Anforderungen																	
1.1	Komplexität		0	3	11	10	0	3	3	10	0	1	4	6	0	7	18	26
1.2	Abhängigkeiten		1	5	6	12	0	1	10	5	0	0	8	3	1	6	24	20
1.3	Quantitative Formulierung	Testbare Formulierung	1	10	10	2	0	4	6	6	0	3	3	5	1	17	19	13
1.4	Priorisierung		3	10	9	2	0	8	5	3	0	3	8	0	3	21	22	5
2	Testbarkeit des Programms																	
2.1	Komplexität		0	4	10	10	1	1	10	4	0	2	2	6	1	7	22	20
2.2	Isolierbarkeit		0	6	10	8	0	1	9	6	0	1	5	5	0	8	24	19
2.3	Trennung der Verantwortlichkeiten		1	7	12	3	0	8	6	1	0	1	8	2	1	16	26	6
2.4	Kontrollierbarkeit		0	2	8	14	0	3	10	3	0	1	5	5	0	6	23	22
2.5	Beobachtbarkeit		4	4	4	11	1	3	6	6	0	1	3	7	5	8	13	24
2.6	Verständlichkeit		2	3	11	7	1	5	8	2	0	2	7	2	3	10	26	11
2.7	Automatisierbarkeit		0	6	10	7	1	4	5	6	0	2	3	6	1	12	18	19
2.8	Diagnostizierbarkeit		5	11	5	1	1	8	3	4	0	4	3	3	6	23	11	8
2.9	Fehler-Lokalisierbarkeit		6	9	4	4	1	5	6	4	0	2	5	4	7	16	15	12
2.10	Robustheit		2	8	8	5	0	5	6	5	0	5	3	3	2	18	17	13
2.11	Heterogenität		1	5	9	9	2	1	8	5	0	5	3	3	3	11	20	17
3	Prozess-Unterstützung																	
3.1	-	Verfügbarkeit der Testreferenz					0	1	6	9	0	0	3	8	0	1	9	17
3.2	Verfolgbarkeit der Testreferenz		0	4	7	13	0	2	7	7	0	2	3	6	0	8	17	26
3.3	Aktualität der Testreferenz		1	5	11	7	0	8	6	2	0	3	5	3	1	16	22	12
3.4	Stabilität der Testreferenz		0	3	8	13	0	4	5	7	0	2	5	4	0	9	18	24
3.5	-	Testumfang					1	7	6	2	0	1	9	1	1	8	15	3
3.6	-	Testdaten					2	5	7	2	0	4	2	5	2	9	9	7

Tabelle 3: Daten der Umfragen

Anhang C Testbarkeitsfaktoren in alternativer Sortierung

Abbildung 7 zeigt die Antworten der Teilnehmer zu den Testbarkeitsfaktoren bzgl. des Programms, **sortiert nach** der Summe der Antwortkategorien „hoch“ und „mittel“.

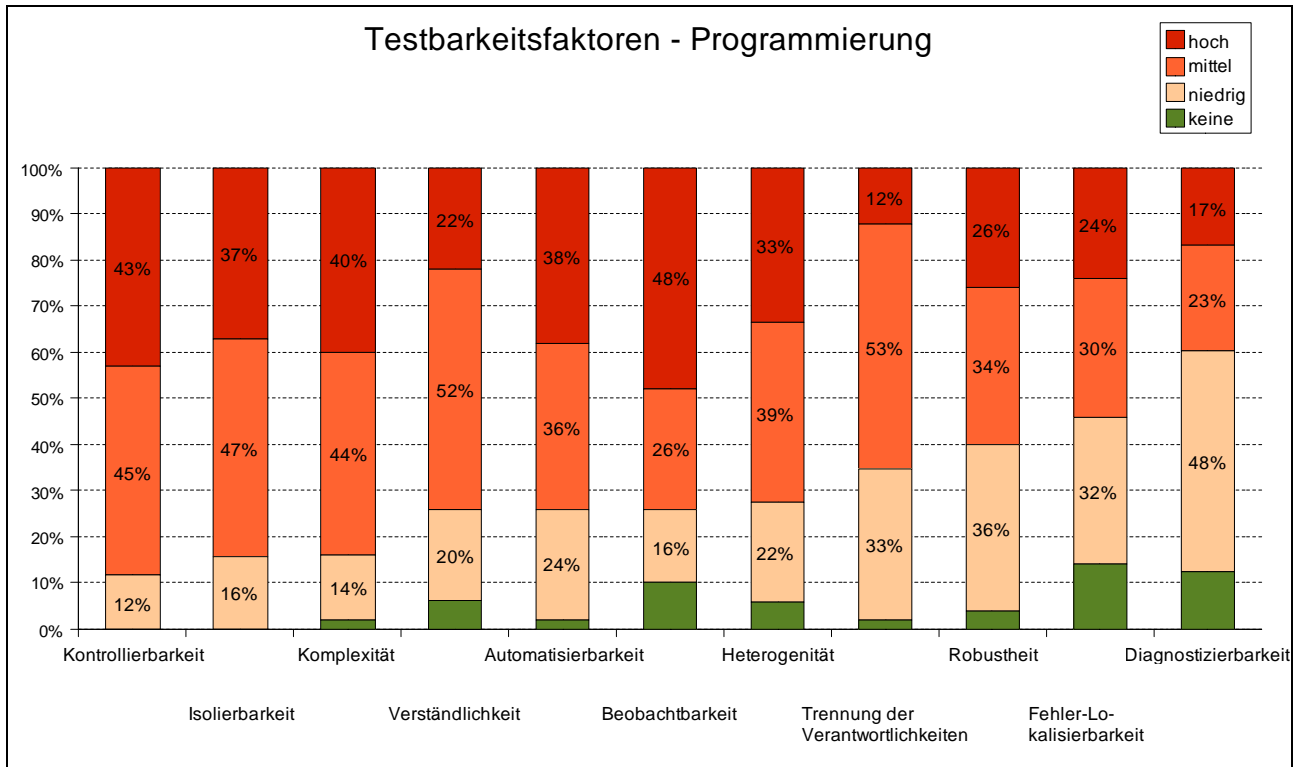


Abbildung 7: Testbarkeitsfaktoren bzgl. des Programms, sortiert nach der Summe der Kategorien „hoch“ und „mittel“.

Anhang D Referenzen

- [Link05] Johannes Link. *Softwaretests mit JUnit: Techniken der testgetriebenen Entwicklung*. 2., überarbeitete und erweiterte Auflage, dpunkt.verlag, Januar 2005, ISBN 3-89864-325-5
- [Jung05] Stefan Jungmayr. *Testbarkeit im Entwicklungsprozess*. In Proceedings des Kolloquiums "Testen im System- und Software-Lifecycle", 29.-30. November 2005, Technische Akademie Esslingen, S. 377-389. ISBN 3-924813-61-2.
- [Spil03] A. Spillner, T. Linz. *Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified-Tester*. dpunkt, 2003.
- [Url:GI] <http://www.gi-ev.de/>
- [Url:TAV] <http://www.gm.fh-koeln.de/~winter/tav/>
- [Url:TOOP] <http://toop.gi-ev.de/>