

Vorgehensmodelle für mobile Anwendungen

Thomas Preisler

INF-M2 - Anwendungen 1
Sommersemester 2008

Department Informatik
HAW Hamburg

20. Mai 2008

Gliederung

- 1 Einleitung
 - Motivation
- 2 Mobiles Testen
 - Grundlagen
 - Black Box Testing
 - White Box Testing
- 3 Mobile Usability
 - Herausforderungen
 - Feld- und Labortests
 - Feld-Tests
- 4 Fazit
- 5 Literatur

Gliederung

- 1 Einleitung
 - Motivation
- 2 Mobiles Testen
 - Grundlagen
 - Black Box Testing
 - White Box Testing
- 3 Mobile Usability
 - Herausforderungen
 - Feld- und Labortests
 - Feld-Tests
- 4 Fazit
- 5 Literatur

Motivation

Für die Entwicklung von immobilen Anwendungen existieren ausgereifte Vorgehensmodelle:

- Testen in Forschung, Lehre und Praxis weitreichend erforscht
 - HAW: WP Certified Tester
- Usability – Untersuchungen ebenfalls in Forschung, Lehre und Praxis weit verbreitet
 - HAW: Usability – Lab

Mobile Anwendungen

Mobile Anwendungen gewinnen immer mehr an Verbreitung und Bedeutung:

- Mobilität bringt neue Herausforderungen im Bereich Testen und Usability
- Vorhandene Vorgehensmodelle decken nicht alle neuen Herausforderungen ab
 - Neue Vorgehensmodelle sind erforderlich (evtl. Abwandlungen)

Gliederung

- 1 Einleitung
 - Motivation
- 2 **Mobiles Testen**
 - Grundlagen
 - Black Box Testing
 - White Box Testing
- 3 Mobile Usability
 - Herausforderungen
 - Feld- und Labortests
 - Feld-Tests
- 4 Fazit
- 5 Literatur

Grundlagen

Mobile Anwendungen werden kaum auf mobilen Geräten entwickelt:

- Entwicklung meist auf leistungsstarken PCs
- Deployment der mobilen Anwendungen auf die mobilen Geräte
 - Diese unterscheiden sich zum Teil sehr stark voneinander

Benutzer erwarten das mobile Anwendungen zuverlässig und stabil sind:

- Anforderung an Stabilität und Zuverlässigkeit oft höher als bei Standard-Anwendungen
- Mobile Anwendungen sind nicht weniger komplex als Standard-Anwendungen

Grundlagen – Schwierigkeiten

Testen *auf* mobilen Geräten ist schwierig:

- Beschränkte Hardware-Ressourcen
- Eingabe meist unkomfortabel
- Beschränkte Ausgabe- und Protokollierungsmöglichkeiten

Testen *von* mobilen Anwendungen ist schwierig:

- Umgebungen der Zielgeräte sind komplex (wireless signals, context-sensitive-information)
 - Konditionen der Umgebung hängen teilweise von Anzahl und Tätigkeiten der Benutzer ab
- Vielfalt der Zielgeräte erschwert die Wiederverwendbarkeit und Wartbarkeit der Testfälle
- Beschränkte Hardware-Ressourcen der Zielgeräte

Grundlagen – Begriffe

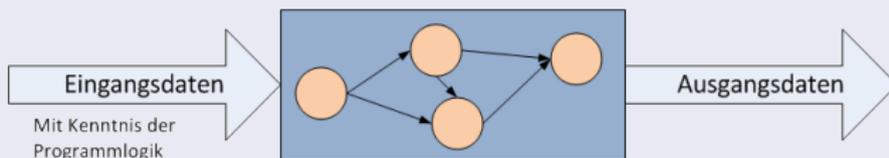
Black Box Test

Funktionaler Test, der das von außen sichtbare Verhalten prüft und dabei keine Kenntnis über die Programmlogik hat.



White Box Test

Strukturbezogener Test, der den Kontrollfluss prüft, Kenntnisse über die Programmlogik und den Quellcode sind vorhanden.



Black Box Testing

MobileTest: Tool für automatisierte Black Box Tests auf mobilen Geräten

- Generiert wartbare/wiederverwendbare Testfälle für Anwendungen auf verschiedenen, mobilen Geräten
- Test-Skripte werden am PC entwickelt
- Mobiler Agent auf Zielgerät führt Test-Fall aus
- PC analysiert und speichert Ergebnisse

Zielsetzung:

- Minimiert die Benutzung von Bilderkennung und Textvergleich zur Statuserkennung
- Bietet Mechanismen um die Umgebung des Zielgerätes zu kontrollieren (Verhalten in unterschiedlichen Umgebungen)

MobileTest: Architektur

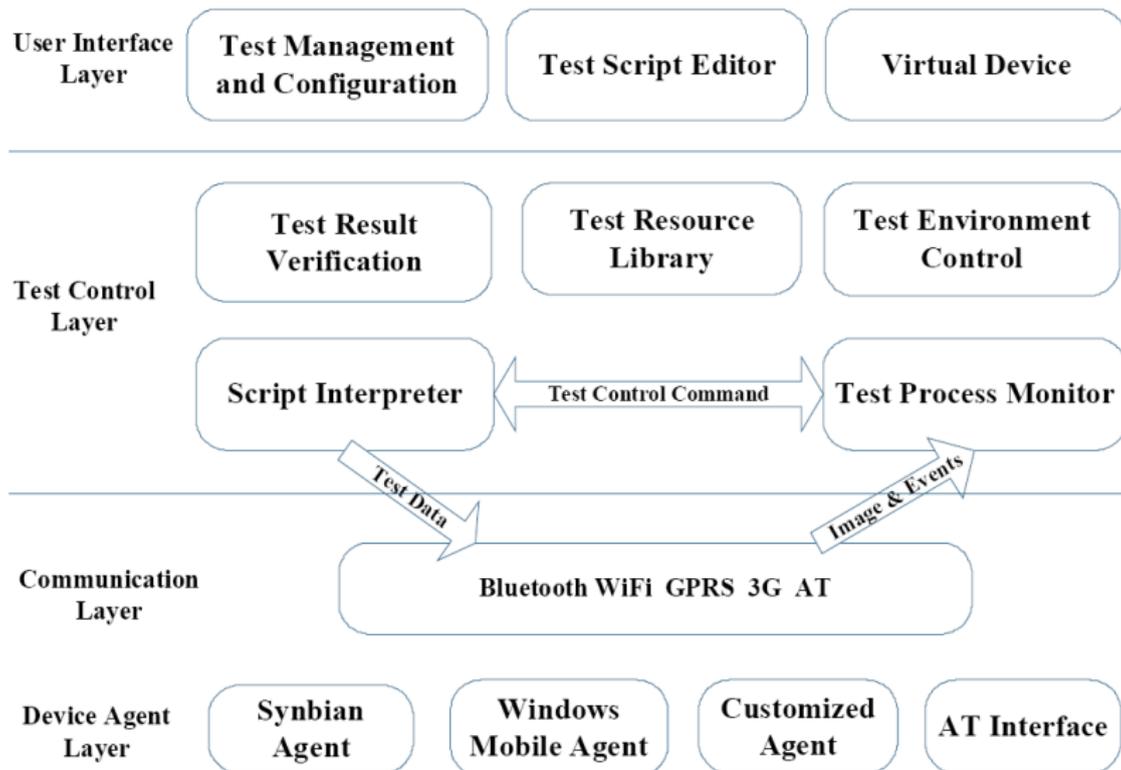


Abbildung: MobileTest Architektur [Bo et al.]

MobileTest: Szenario

- 1 Konfiguration der Testumgebung und des Zielgerätes
- 2 Erstellen der Testskripte (manuell oder automatisch)
- 3 Skripte werden ausgeführt
- 4 Skript Interpreter sendet Input zum Device Agent
- 5 Device Agent simuliert die Eingaben auf dem Zielgerät und sendet Ergebnisse an den Test Monitor
- 6 Test Monitor speichert Ergebnisse in der Resource Library
- 7 Ergebnisauswertung

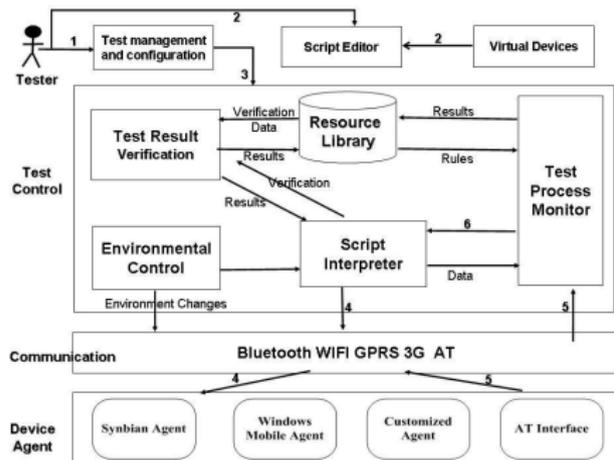


Abbildung: MobileTest Szenario [Bo et al.]

White Box Testing

Status quo:

- Es existieren kaum Ansätze um die Korrektheit einer mobilen Anwendung zu testen
- Aber: mobile Anwendungen müssen wie alle anderen getestet werden

JaBUTi/ME: Tool für White Box Tests auf mobilen Geräten (Java)

- Überprüft die Korrektheit einer Anwendung
- Teststrukturen leiten sich aus der Implementierung ab (Statements, Entscheidungen, Datenfluss, Interaktionen, ...)
- Analysiert wird der Byte-Code nicht der Quellcode
- Anwendungen auf den Zielgeräten getestet (nicht im Emulator)

JaBUTi/ME in action

Control Flow Graph

Abstrakte Darstellung eines Programmes bzw. eines Programmteils.

- Ein gerichteter Graph
 - Knoten repräsentieren unteilbare Code-Blöcke
 - Kanten repräsentieren mögliche Ablaufsequenz zwischen den Blöcken
- 1 Analyse des Programm-Codes und CFG erstellen
 - 2 Test-Anforderungen, die erfüllt sein müssen, werden generiert
 - 3 Code-Instrumentalisierung (Code wird erweitert):
 - Informationen welcher Block ausgeführt wird
 - Sammeln, speichern der Daten
 - 4 Analyse, welche Test-Anforderungen durch die Test-Fälle abgedeckt werden

JaBUTi/ME: Server-based testing

- Mobile Geräte: zu begrenzte Ressourcen für eine Testumgebung
- Aber: Wichtig, dass in der realen Umgebung getestet wird
- Daher: Anwendung wird auf mobilem Gerät getestet und Ausführungsdaten über das Netzwerk versendet
- Server (Desktop-PC) empfängt diese und wertet sie aus

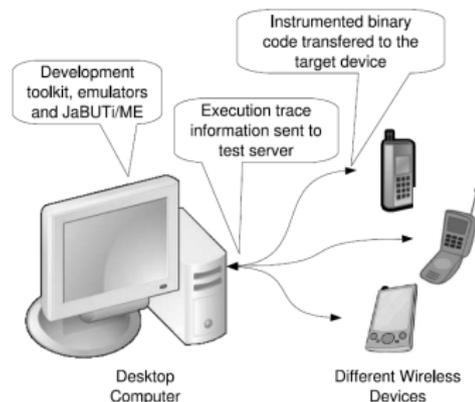


Abbildung: JaBUTi/ME: Server-based testing [Delamaro et al.]

Gliederung

- 1 Einleitung
 - Motivation
- 2 Mobiles Testen
 - Grundlagen
 - Black Box Testing
 - White Box Testing
- 3 Mobile Usability**
 - Herausforderungen
 - Feld- und Labortests
 - Feld-Tests
- 4 Fazit
- 5 Literatur

Probleme/Herausforderungen

Infrastruktur Herausforderungen:

- Benötigte Zeit für Aufbau, Konfiguration der Geräte
- Anzahl der Beobachter
- Kompatibilität mit vorhandener Infrastruktur (immobile Geräte)

Herausforderungen von mobilen Geräten:

- Gewicht der Kamera/des Mikrofons
- Konsequenzen, die aus einer evtl. Beschädigung der Geräte entstehen
- Soziale Komponenten durch mobilen Einsatz

Herausforderungen der Usability–Untersuchung:

- Dauer der Usability–Untersuchung
- Audio- und Videoaufnahmen
- Live Überwachung der Untersuchung

Feld- oder Labortests?

- Konventionelle Usability–Untersuchungen meist im Labor
- Mobile Anwendungen werden aber fast immer unterwegs benutzt

Pro Labor–Test:

- Weniger zeitaufwändig
- Sammeln und speichern der Daten einfacher
- Vereinfacht die Kontrolle der Untersuchung

Pro Feld–Test:

- Benutzerverhalten im Labortest unterscheidet sich vom Verhalten in der realen Situation
- Unkontrollierbare Umweltfaktoren beeinflussen die Anwendung
 - Im Feld–Test werden signifikant mehr Usability–Fehler gefunden
 - Benutzer benötigen für Aufgaben im Feld–Test länger

[Nielson et al.]

Infrastruktur für Feld-Tests

Schwierigkeiten:

- Wie wird der Bildschirm des mobilen Gerätes aufgezeichnet?
- Wie werden die Eingaben des Benutzers aufgezeichnet?
- Wie werden die Reaktionen des Benutzers aufgezeichnet?
- Wie/Wo werden die gewonnenen Daten unterwegs gespeichert?

Lösungen:

- Ein Kameramann filmt den Probanden
 - Privatsphäre? Braucht man nicht eher drei Kameramänner?
- An dem Gerät werden mobile Kameras montiert
 - Wird die Bedienung des Gerätes dadurch eingeschränkt?
- Mobiles Gerät überträgt Daten über das Internet an einen Server (teuer)
- Mobiles Gerät überträgt Daten per Funk an einen Laptop in Reichweite

Infrastruktur für Feld-Tests II



Abbildung: Mini-Kamera mit Barcode Scanner [Nielsen et al.]



Abbildung: Observation System [Schusteritsch et al.]

Gliederung

- 1 Einleitung
 - Motivation
- 2 Mobiles Testen
 - Grundlagen
 - Black Box Testing
 - White Box Testing
- 3 Mobile Usability
 - Herausforderungen
 - Feld- und Labortests
 - Feld-Tests
- 4 **Fazit**
- 5 Literatur

Fazit

Testen:

- Als Disziplin in der Theorie weitreichend erforscht
- Viele interessante/erfolgsversprechende Ansätze bereits vorhanden
- Allerdings fehlt es noch an einer Übersichtsarbeit
- *Fazit*: Vergleich, Evaluierung der Ansätze wäre interessant (Übersichtsarbeit)

Usability:

- Als Disziplin in der Theorie weitreichend erforscht
- Für immobile Anwendungen Infrastruktur vorhanden
- Für mobile Anwendungen wird noch an passender Infrastruktur experimentiert
- *Fazit*: Bereich der Usability für mobile Anwendungen noch nicht komplett erschlossen

Literatur



Robert V. Binder, James E. Hanlon.

The advanced mobile application testing environment.

SIGSOFT Softw. Eng. Notes, 30(4):1, 2005.



Jiang Bo, Long Xiang, Gao Xiaopeng.

MobileTest: A Tool Supporting Automatic Black Box Test for Software on Smart Mobile Devices.

AST '07: Proceedings of the Second International Workshop on Automation of Software Test, 2007.



M. Cundy.

Testing Mobile Applications is Different from Testing Traditional Applications.

Veritest Tester's Network, 2001.



M. E. Delamaro, A. M. R. Vincenzi, J. C. Maldonado.

A strategy to perform coverage testing of mobile applications.

AST '06: Proceedings of the 2006 international workshop on Automation of software test, 2006.



H. Been-Lirn Duh, G. C. B. Tan, V. Hsueh-hua Chen.

Usability evaluation for mobile device: a comparison of laboratory and field tests.

MobileHCI '06: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services, 2006.

Literatur II



Kwang Book Lee, Roger A. Grice.

Developing a New Usability Testing Method for Mobile Devices.

Professional Communication Conference, 2004. IPCC 2004. Proceedings. International, 2004.



Qusay H. Mahmoud.

Testing Wireless Java Applications.

Sun Developer Network, 2002.



Christian Monrad Nielsen, Michael Overgaard, Michael Bach Pedersen, Jan Stage, Sigge Stenild.

It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field.

NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction, 2006.



Rudy Schusteritsch, Carolyn Y. Wei, Mark LaRosa.

Towards the perfect infrastructure for usability testing on mobile devices.

CHI '07: CHI '07 extended abstracts on Human factors in computing systems, 2007.

Vielen Dank für die Aufmerksamkeit...

...noch Fragen?