



Besonderheiten der Anwendungsentwicklung für die mobile Nutzung – ein Leitfaden, Teil 5.

Die Autoren



Dipl.-Ing. Stefanus Römer arbeitet als Produktmanager bei T-Mobile, wo er insbesondere für das Produkt Mobile IP VPN und für mobile Intranet-Access-Lösungen zuständig ist.



Dipl.-Ing. Marcus Freitag studierte an der Universität Koblenz-Landau Informatik mit den Schwerpunkten Rechnernetze und Wirtschaftsinformatik. Seit Juli 2004 arbeitet er als Senior Consultant in der DIALOGS Software GmbH in München.

Der in Heft 11/2004 begonnene Beitrag wird mit dem fünften und letzten Teil zum Thema Testen fortgesetzt. In der Phase des Testens wird die Implementierung entsprechend der bestehenden Anforderungen überprüft. Dabei wird die erstellte Anwendung mehreren systematischen Testabschnitten mit verschiedenen Schwerpunkten unterzogen. Am Ende der Testphase steht der Pilottest mit echten Benutzern in einer realen Umgebung.

Einführung

Die Qualitätssicherung ist eine wichtige Aufgabe in jedem Software-Entwicklungsprojekt, die nicht erst mit Abschluss der Implementierung beginnt, sondern in allen Entwicklungsphasen durchgängig gewährleistet sein muss. In den frühen 80er Jahren sah man die Aufgabe der Qualitätssicherung lediglich als eine Phase am Ende der Implementierung an. Das Testen der Software wurde in der Regel von den Entwicklern selbst erledigt, um noch die „letzten“ Fehler aus der Software zu entfernen. Heute hat sich die Ansicht durchgesetzt, dass sich das Testen über alle Phasen

mit jeweils unterschiedlicher Zielsetzung erstreckt (Bild 23) und von unabhängigem Personal durchgeführt werden sollte. Diese geänderte Philosophie liegt bereits dem inkrementellen Phasenmodell der Software-Entwicklung (s. Bild 2) zu Grunde.

In allen Phasen des Entwicklungsablaufs wird begleitend eine Qualitätssicherung durchgeführt. Die Vorbereitung und die Durchführung des Testens selbst finden ebenfalls in aufeinander folgenden Stufen statt (Bild 24). Zur Durchführung der Tests sollten geeignete Testwerkzeuge in Form von Testsoftware (so genannter Testware) erstellt werden. Das

Das Thema im Überblick

Die Entwicklung von Software für die mobile Nutzung ermöglicht es dem Kunden, seine Kommunikations-Einrichtungen mit größtmöglicher Flexibilität an fast jedem Ort anzuwenden. Damit dies gewährleistet ist, wird jede neu entwickelte Software ausgiebig auf ihre Funktion und ihr Verhalten unter realen Einsatzbedingungen getestet, bevor sie in den Wirkbetrieb überführt werden kann. Hierzu werden im vorliegenden Beitrag Tipps und Empfehlungen gegeben, die einen reibungslosen Testablauf gewährleisten.

weit mehr Test- und Debugging-Zyklen als bei anderen Software-Projekten.

Im folgenden Abschnitt werden aufeinander aufbauende Testphasen beschrieben und praktische Hinweise für die Testdurchführung gegeben.

Testen mobiler Anwendungen

Grundsätzlich sollten beim Test mobiler Anwendungen allgemein bekannte Testgrundsätze und -techniken berücksichtigt werden. Beispielsweise sollten während der Implementierungsphase regelmäßig Modul- und Integrationstests sowie White-Box- und Black-Box-Tests durchgeführt werden³⁹. Weiterhin ist es ratsam, die Tests von unabhängigen Testern vornehmen zu lassen.

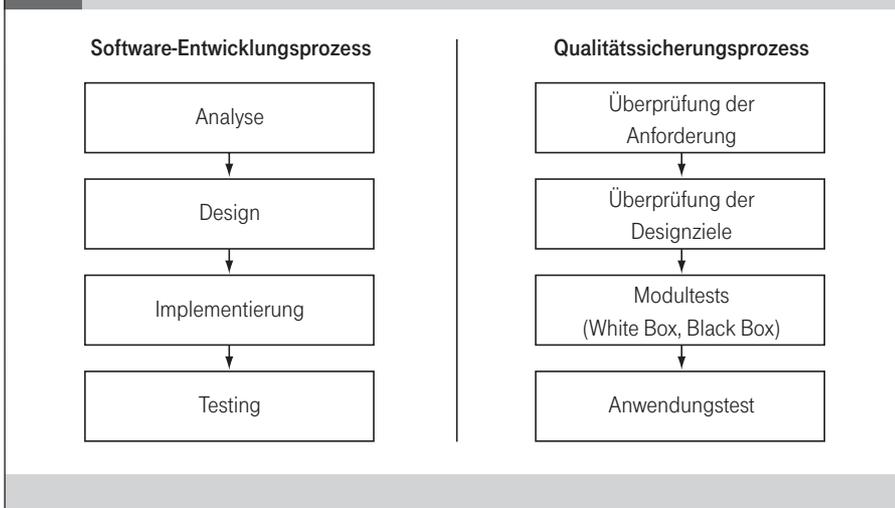
Am Ende der Implementierungsphase liegt die erste Gesamtversion vor, die Schritt für Schritt bis hin zu einem Pilottest außerhalb der Entwicklungsumgebung getestet werden kann. Hierbei lassen sich allgemein folgende Stufen unterscheiden (Bild 25):

■ Testvorbereitung (s. Bild 24)

Erstellung von Testwerkzeugen und

³⁹ Eine vollständige Software-Prüfung erfordert Tests jeweils mit einer White Box und einer Black Box. Die **White Box** wird auch „Gläserner Kasten“ genannt und verlangt vom Tester, genau zu wissen, wie der Programmcode abläuft und was die Software bewirken soll, um zu erkennen, ob das Programm von seinem eigentlichen Ziel abweicht. Beim **Black-Box**-Test ist es nur wichtig zu wissen, wie das Eingangssignal und das erwartete Ausgangssignal auszusehen hat und nicht, wie das Programm zu diesem Ergebnis gelangt.

Bild 23 Qualitätssicherung als begleitende Aktivität während des gesamten Projektes



Vorgehensmodell leitet sich ebenfalls aus dem allgemeinen Phasenmodell der Software-Entwicklung (s. Bild 1) ab.

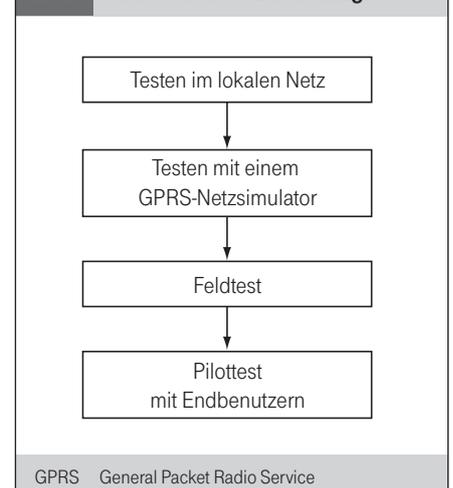
Jede Entwicklungsphase besitzt ein eigenes Testziel, wie es in Bild 23 dargestellt ist. Die

folgenden Abschnitte konzentrieren sich auf allgemein gültige Techniken und Teststrategien für die abschließende Phase des Anwendungstests, weil sich die Testanforderungen in jedem Software-Projekt stark unterscheiden.

Die Phase des Anwendungstests beginnt nach Abschluss der Implementierung. Häufig ist das Ergebnis des ersten so genannten „Alpha“-Tests unbefriedigend und kann die Qualitätsanforderungen bezüglich der Schnelligkeit und der Stabilität des Programmablaufs nicht erfüllen. Ein Grund hierfür liegt darin, dass die meisten Anwendungen unter der Annahme einer Umgebung mit idealen Bedingungen entwickelt werden. In vielen Fällen werden die ersten Tests von den Entwicklern selbst durchgeführt, die davon ausgehen, dass ihre Software weitgehend „fehlerfrei“ ist.

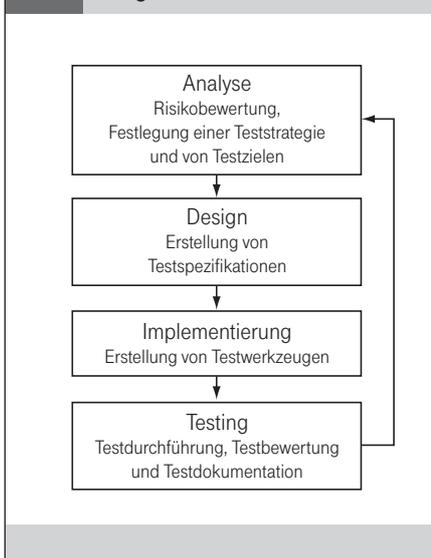
Die Leistungsdaten und die Stabilität mobiler Anwendungen hängen von vielen externen Faktoren ab, die häufig noch unkalkulierbar sind. Daher erfordert die Testphase einer Software-Entwicklung für die mobile Nutzung

Bild 25 Phasen des Anwendungstests



GPRS General Packet Radio Service

Bild 24 Testplanung und Testdurchführung in aufeinander folgenden Phasen



Bereitstellung einer geeigneten Testumgebung.

■ Testen im lokalen Netz

Zu Beginn müssen erste Funktionstests in einem lokalen Netz durchgeführt werden. Hierzu werden die unten beschriebenen Testwerkzeuge benötigt.

■ Testen mit einem GPRS-Netzsimulator

Ein GPRS-Netzsimulator soll möglichst realistische Testbedingungen nachbilden. Die Tests, die zuvor im lokalen Netz durchgeführt wurden, werden nun mit Hilfe eines GPRS-Netzsimulators erneut durchlaufen.

■ Feldtest

Ein Feldtest wird durchgeführt, nachdem die Tests im simulierten Netz abgeschlossen sind. Dabei werden gezielt Grenzsituationen im realen Mobilfunknetz gesucht, um das Verhalten der Anwendung erneut zu überprüfen.

■ Pilottest mit realen Benutzern

Im Pilottest soll sich die Benutzerfreundlichkeit und Anwendbarkeit (Usability) der Software abschließend als funktionsfähig und zufrieden stellend erweisen.

Testwerkzeuge

Es wird empfohlen spezielle Testwerkzeuge zu erstellen. Der Einsatz von Testwerkzeugen erlaubt ein eigenständiges Testen von einzelnen Anwendungsteilen und vereinfacht somit die Testdurchführung und die Fehlereingrenzung. Die Entwicklung dieser Werkzeuge erfordert einen zusätzlichen Aufwand, verkürzt allerdings anschließend die Test- sowie Debugging-Zyklen. Bereits vorhandene Software-Teile der Client- und der Server-Implementierung können bei der Entwicklung der Testwerkzeuge genutzt werden. Dies hat einen entscheidenden Vorteil: Die Fehler, die bei der Entwicklung der Testwerkzeuge gefunden und beseitigt werden, stellen gleichzeitig Fehler in der Anwendungs-Software dar, die somit ebenfalls behoben werden können. Im Folgenden werden die Anforderungen an verschiedene Testwerkzeuge beschrieben. Weitergehende Einzelheiten hängen im Wesentlichen vom jeweiligen Software-Projekt ab und können daher nicht allgemein beschrieben werden.

Client-Simulator

Ein Client-Simulator bildet eine Client-Sitzung mit realistischen Transaktionen nach. Neben der Überprüfung der Implementierung des Sitzungs-(Session-)Managements sowie des Protokolls als Ganzes erlaubt dieses Verfahren die Kontrolle jeder einzelnen Aktion, wie beispielsweise die An- und Abmeldung oder die Bearbeitungszeit bestimmter Transaktionen. Ein Client-Simulator sollte folgende Leistungsmerkmale bieten:

- Simulieren einer Client-Sitzung,
- An- und Abmelden beim Server,
- Senden und Empfangen simulierter Daten während einer Sitzung,
- Überprüfen der Antworten auf Korrektheit,
- Messen von Bearbeitungs- und Antwortzeiten sowie
- Versenden fehlerhafter Daten.

Folgende Testziele werden verfolgt:

- Überprüfen der Implementierung und des Protokolls,
- Messen von Leistungsparametern wie z. B. Antwortzeiten und
- Überprüfen der Server-Stabilität im Falle fehlerhafter Daten.

Multi-Client-Simulator

Ein Multi-Client-Simulator wird zur Nachbildung mehrerer zeitlich paralleler Client-Sitzungen benötigt. Mit einem Multi-Client-Simulator überprüft der Tester die Funktion und die Stabilität des Client-Managements innerhalb des Servers. Im Mittelpunkt steht hierbei die Überprüfung der Implementierung von Threads und deren Zugriff auf gemeinsame Ressourcen auf der Server-Seite.

Ein Multi-Client-Simulator sollte eine beliebig einstellbare Anzahl paralleler Sitzungen simulieren können, wobei folgende Testziele verfolgt werden:

- Überprüfen des Multi-Client-Managements auf der Server-Seite und
- Erkennen von Blockierungen (Deadlocks), Zugriffskonflikten und Stabilitätsproblemen.

Last-Simulator

Der Last-Simulator erzeugt eine möglichst realistische Nutzlast auf Seiten des Servers. Er wird benötigt, um die Belastungsgrenzen des Servers zu testen und herauszufinden, wie der Server bei Überlast reagiert. Da ein Lasttest eine sehr große Datenmenge erzeugt, sollte er nur in einem lokalen Netz mit entsprechend hoher Bandbreite durchgeführt werden. Ein Last-Simulator sollte folgende Anforderungen erfüllen:

- Simulieren der maximal zulässigen Anzahl paralleler Client-Sitzungen,
- Simulieren der maximalen Anzahl gleichzeitiger Transaktionen und
- Durchführen von Langzeittests über mehrere Tage oder Wochen.

Mit einem Lastsimulator werden folgende Testziele verfolgt:

- Überprüfen von Leistungsdaten und Stabilität in einer realistischen Situation, bei Dauer- sowie Spitzenlast und
- Erkennen von Ressourcen-Engpässen (z. B. Speicher, Prozessor) und Instabilitäten.

Testen im lokalen Netz

Die ersten Tests des Gesamtsystems werden in der lokalen Entwicklungsumgebung durchgeführt. Die nahezu „idealen“ Netzbedingungen erlauben es, sich zunächst auf die Anwendungslogik und die Funktion der Kommunikations-Komponente zu konzentrieren. Die Durchführung von Lasttests ist ebenfalls nur in einem lokalen Netz empfehlenswert. Erst nachdem die Software in dieser Umgebung stabil läuft, empfiehlt sich ein Wechsel der Testumgebung. Die dazu erforderlichen Testschritte sind:

- Überprüfen der Client-Implementierung mit Hilfe des Client-Simulators und Fehlerbeseitigung in der Client-Implementierung,
- Beseitigen von Fehlern im Transaktionsprotokoll und im Session-Management,
- Überprüfen der Login- und der Logout-Prozedur auf der Server-Seite mit Hilfe des Client-Simulators,

- Simulieren realistischer Transaktionen mit Hilfe des Client-Simulators,
- Simulieren mehrerer paralleler Sessions mit Hilfe des Multi-Client-Simulators,
- Lasttest im Dauerbetrieb und
- Ende-zu-Ende-Test mit Hilfe der Client-Applikation.

Testen mit einem GPRS-Netzsimulator

Nachdem die ersten Fehler in einer lokalen Netzumgebung gefunden und beseitigt worden sind, kann die Anwendung mit Hilfe eines GPRS-Netzsimulators getestet werden. Ein GPRS-Netzsimulator erlaubt ein „realitätsnahes“ Testen ohne Verwendung eines mobilen Endgerätes. Dadurch kann das Verhalten der Anwendungslogik und des Session-Managements ohne Einflüsse von Hardware- oder Firmware-Fehlern getestet werden.

Die Bereitstellung eines GPRS-Netzsimulators ist relativ einfach. Durch den Einsatz eines zusätzlichen Linux-Servers als Standard-Gateway für Client- und Server-Applikationen im lokalen Netz lassen sich die Paketlaufzeiten und die Paketverlustrate künstlich verschlechtern. Darüber hinaus sind verschiedene allgemein zugängliche Simulatoren kostenfrei im Internet abrufbar.

Wird ein GPRS-Netzsimulator verwendet, werden die gleichen Testschritte wiederholt wie in „idealer“ Testumgebung ohne Netzsimulator.

Feldtest

In der Phase des Feldtests wird zum ersten Mal die gesamte Anwendung einschließlich der vollständigen Endgeräteplattform eingesetzt und getestet. Die Fehler, die leicht zu finden waren, wurden bereits in den vorangegangenen Phasen beseitigt. Im Feldtest soll insbesondere das Verhalten der Client-Plattform und der Kommunikationskomponente analysiert werden⁴⁰. Zu diesem Zweck wer-

den realistische Nutzungssituationen nachgestellt (z. B. ein Telefonat während einer Autofahrt mit aktiver GPRS-Verbindung) und gezielt Grenzsituationen im realen GPRS-/UMTS-Netz gesucht. In dieser Phase sollten Endbenutzer noch nicht teilnehmen.

Die erforderlichen Testschritte sind:

- Prüfen allgemeiner Bedienbarkeit und des Verhaltens der Gesamtanwendung,
- Nachstellen typischer Anwendungssituationen: stationäre und mobile Nutzung (z. B. im Auto oder im Zug),
- Nachstellen von Grenzsituationen wie beispielsweise: schwache Netzversorgung, Übergang in ein Gebiet ohne Netzversorgung und Rückkehr in ein versorgtes Gebiet, Abschirmung der mobilen Empfangseinheit, Verbindungsunterbrechung, Akkuentladung und
- Analyse des Stromverbrauchs.

Die direkte Unterstützung durch den Mobilfunk-Netzbetreiber, beispielsweise durch T-Mobile, ist in dieser Phase und während des anschließenden Pilottests von besonderer Bedeutung. Der Netzbetreiber kann bei Bedarf Kommunikationsabläufe innerhalb des Mobilfunknetzes gezielt mitprotokollieren, um damit Fehlerzustände schneller zu analysieren.

Pilottest

Der Übergang von der Phase des Feldtests zum Pilottest zeichnet sich dadurch aus, dass nun zum ersten Mal Endbenutzer einbezogen werden. Ziel ist es dabei, die Akzeptanz der Nutzer und das Verhalten in realen Nutzungssituationen zu analysieren. In dieser Phase findet deshalb ein eingeschränkter Rollout⁴¹ mit einer repräsentativen Auswahl von Benutzern statt.

Schlussbetrachtung

Im vorliegenden Beitrag wurden grundlegende Testmethoden und -strategien vorgestellt, die ein systematisches Testen mobiler Anwendungen unter Berücksichtigung der besonderen Eigenschaften der mobilen Daten-

kommunikation ermöglichen. Der Application Configuration & Developer Guide (ACDG) der T-Mobile stellt dem Anwendungsentwickler und Netzadministrator eine umfangreiche Wissenssammlung bereit, die es ihm ermöglicht, Entwicklungsprojekte für mobile Anwendungen zum Erfolg zu führen. Im ACDG werden die besonderen Eigenschaften mobiler Übertragungsmedien und deren Auswirkungen auf die Anwendungsentwicklung anhand von konkreten Codebeispielen eingehend erläutert.

Die mit dem fünften Teil abgeschlossene Beitragsserie bietet somit eine Einführung in das Grundlagenwissen der Anwendungsentwicklung für mobile Nutzung. Die Berücksichtigung der hier beschriebenen Empfehlungen in einer frühen Entwicklungsphase hilft, Aufwand und Kosten zu sparen, Zeit zu gewinnen und Fehleinschätzungen zu vermeiden. (Br)

Internetadresse:

Kostenfrei allgemein zugängliche Netzsimulatoren sind im Internet unter http://info.iet.unipi.it/~luigi/ip_dummyenet/ oder <http://www.knowplace.org/shaper/> abrufbar.

Literaturhinweis:

Application Configuration & Developer Guide, T-Mobile International 2004, www.t-mobile.de/entwickler.

⁴⁰ Siehe hierzu den Beitrag „Besonderheiten der Anwendungsentwicklung für die mobile Nutzung – ein Leitfaden, Teil 2“, WissenHeute 12/2004, S. 660 ff.

⁴¹ **Rollout:** Zeitpunkt, an dem eine Neuentwicklung in den Wirkbetrieb überführt wird.