

Neue Betriebssysteme und Anwendungsplattformen

Umbruch (auch) für die Agrarinformatik

Referat zur 32. Jahrestagung der Gesellschaft für
Informatik in der Land-, Forst- und Ernährungswirtschaft
(Weihenstephan, 29.02. – 01.03.2012)

Hagen F. Piotraschke, Software- und Systementwickler
OptimalSystem.DE, D-04668 Grimma
hagen@piotraschke.de

1990er – ca. 2010: Heile Welt ☺

Laufzeitumgebungen in fest gefügter Ordnung

Windows-PC

- quasi 100%ige Verbreitung
- extreme Abwärtskompatibilität
- System-APIs (fast) völlig offen
- Runtimes für jeden Geschmack

PDA

- Palm OS → Windows Mobile
- lokaler Sync mit Windows PC

Maschinen-Terminals

- Black Boxes (OS, Runtime, DB, App, UI ...)
- RS232, CAN, FAT, ... aber keine APIs/ABIs

Webserver

- UNIX (→ Linux) oder Win Server
- OS, Runtime und DB abgeschirmt
- HTML (+ Javascript), SOAP, REST ...

Schöne neue Welt !?

Reorganisation in der Post-PC-Ära

Steve Jobs (2010): *„I’m trying to think of a good analogy. When we were an agrarian nation, all cars were trucks. But as people moved more towards urban centers, people started to get into cars. I think PCs are going to be like trucks. Less people will need them. And this transformation is going to make some people uneasy... because the PC has taken us a long way. They were amazing. But it changes. Vested interests are going to change. And, I think we’ve embarked on that change. Is it the iPad? Who knows? Will it be next year or five years? ... We like to talk about the post-PC era, but when it really starts to happen, it’s uncomfortable.“*

Revolution von oben

- Apple, Google, Microsoft, Amazon
- Samsung, Motorola, Nokia ...
- Patentkrieg → Verdrängung(en)?
- Zwangsbindung an Cloud-Dienste
- Abschottung auf Geräte-Ebene

Revolution von unten

- Kaufverhalten, Gewohnheiten der Nutzung, *Bring Your Own Device*
- Bindung Person an Systemanbieter
- Komfort- und Sicherheitsniveau
- IT-Geräte als Statussymbole

Herausforderung für angewandte IT

Anwendungen und Dienste für die neuen Plattformen

Anforderungen aus Nutzerperspektive

- Bereitstellung der gewünschten Fachanwendung für vorhandene Geräte bzw. Plattformen (alternativ: Gerätekauf eigens für spezielle Anwendungen)
- GUI bzw. „Look & Feel“ sollte den Standards der neuen Plattformen folgen
- Erwartungshaltung aus zunehmender Gewöhnung an Cloud-Services
- Bequemlichkeit steht an erster Stelle – Sicherheit ist selbstverständlich? 😊

Probleme für Entwickler und Systemanbieter

- Daumenschrauben der Plattformanbieter (erforderliche Programmiersprachen und -Umgebungen, Vorgaben für App Stores, Abschottungen der Geräte für lokale Schnittstellen und bisher übliche Techniken) * Anzahl der Plattformen
- vorhandener Quellcode kaum weiterverwendbar (Schichtenarchitekturen?)
- kurzfristige Umstellung auf völlig neue GUI-Techniken (z.B. Touchscreens)
- Interessenkonflikt: Sicherheit sollte wichtiger sein als Anwenderkomfort! ☹️

Lösungswege in der Praxis

Strategien & Kompromisse unter Ressourcenknappheit

Weiterführung bzw. Adaption etablierter Anwendungen/Dienste

- Virtualisierung (z.B. Bereitstellung von PC-Programmen auf Citrix-Servern)
- Bereitstellung und Anbindung von Cloud-Services für bestehende Installationen (Speicherplatz für Nutzerdaten auf Server, Sync mit alternativen Apps)
- Ergänzung durch Apps mit wichtigen bzw. häufig genutzten Teilfunktionen
- Ergänzung oder Substitution durch Webanwendungen (wenn geeignet)

Entwurf und Bereitstellung neuer Systeme

- Systemarchitektur im Entwurf immer „top-down“ (Server → Clients)
- Möglichkeit der lokalen Datenhoheit des Anwenders (wenn gewünscht) prüfen
- Entwicklung für verschiedene Plattformen nach Markt- bzw. Bedarfsanalyse sowie nach technischer Eignung (*cross platform tools?*) und eigenem Personal
- möglichst strikte Schichtentrennung (Geschäftslogik wiederverwendbar?)
- möglichst unabhängig von Plattform-Versionen, Bildschirmauflösung usw.

Hürden, Stolpersteine & Fesseln

Probleme und Einschränkungen für die Entwickler

- Fragwürdigkeit von Hersteller-Aussagen zur Zukunftssicherheit von Plattformen, Entwicklungstechniken usw. (z.B.: .NET und Java als Heilsversprechen auf langfristige Plattformunabhängigkeit bzw. Flash und Silverlight als dauerhaft optimale Synthese von Web- u. lokaler Anwendung)
- Workarounds für Schnittstellen- und Protokollprobleme z.T. sehr schwierig (z.B. Anbindung von GNSS, Dateisysteme, Peripherie via USB, CAN o. RS232)
- Beibehaltung obsoleter Software-Techniken wirkt offenkundig antiquiert (Aussehen und Bedienbarkeit der GUI-Elemente, Laufzeitverhalten usw.), stark veraltete Techniken wie VB6 jederzeit vom „Aussterben“ bedroht
- Schnelle Reaktionen auf Daten- oder Programmfehler nur serverseitig möglich (Durchlaufzeiten bei App Stores, ggfs. kein Direktzugriff am Gerät)
- C/C++ (WinRT/WinAPI), C#/.NET, Java und Objective-C relativ inkompatibel
- Keine klaren Roadmaps für wichtige Leistungsmerkmale (z.B. SPP für Windows Phone, Multi-Threading für Mobilplattformen auf ARM-Basis)

Beispiel: *FeldBild*-System

Server- und Client-Software für Themenkarten & Geobilder

Problemstellung für Systemarchitektur

- Datenmengen bzw. Einzeldateigrößen ungeeignet für Webanwendung
- Zugriff auf lokale Schnittstellen bzw. Dateisysteme (USB-Anschluss für Kameras und GNSS-Geräte bzw. Massenspeicher wie SD-Karten usw.) erforderlich
- Notwendigkeit der Arbeitsteilung zwischen Desktop- und Mobilanwendung
- GNSS (besser DGNSS oder PPP) notwendig für Mobilanwendung(en)
- Begrenzung der Entwicklungskapazitäten → kleines Ingenieurbüro 😊

Lösungsansatz und Realisierungsabfolge

- Serveranwendung (zentrale Datenbank, Sync, Web-UI für Teilaufgaben)
- PC-Anwendung für Windows (.NET/WPF) mit konventionellem GUI (Menü usw.)
 - von Beginn an auf Server-Sync in Background Threads hin entwickelt
- Mobile Apps in HTML5 für Teilaufgaben (nur eingeschränkt verwendbar)
- Native Apps nach Möglichkeit (bevorzugt Windows 8 & Windows Phone 7 ff.)

Zusammenfassung & Ausblick

Mehr Probleme und Risiken als Chancen?

Positiva

- höhere Anwendungssicherheit auf Geräteebene → geringerer Aufwand für Support je Anwender → höhere Skalierbarkeit (bei Agrarsoftware wichtig?)
- Apple, Google u. Microsoft als „Türöffner“ für neue Cloud-Services von KMU

(Weitere) Negative

- Preisniveau und Kaufgewohnheiten in App Stores (In-App-Käufe, Sync-Abos?)
- Fehlende Interoperabilität auf Geräteebene, Abschottungsproblematik
- Entwickler bzw. KMU-Systemanbieter sind ab jetzt schwächstes Glied der Kette

Fazit

- Festhalten am Bisherigen ist aus Sicht von Entwicklern und Systemanbietern für bestehende Anwendungen zunächst ggfs. (noch) wirtschaftlich völlig vernünftig
- Spezialisierung auf eine einzige Plattform könnte sinnvoll sein (bzw. bleiben)

Vielen Dank für Ihre Aufmerksamkeit!

Weiterführende Literatur

- Zittrain, Jonathan (2011-11-30): „*The Personal Computer Is Dead*“
<http://futureoftheinternet.org/the-pc-is-dead-why-no-angry-nerds>
<http://www.heise.de/tr/artikel/Wir-brauchen-wuetende-Nerds-1397391.html>
- Jaser, Michael (2011-03-21): „*Evaluation, Bewertung und Implementierung verschiedener Cross-Platform Development Ansätze für Mobile Internet Devices auf Basis von Web-Technologien*“ (Hochschule Augsburg, Fakultät für Informatik)
<http://cross-mobile-apps.de/files/bachelorthesis-michael-jaser.pdf>