# Enduser-Development

Johannes Böttcher
Institut für Informatik
FU Berlin
11.01.2007

# Gliederung



- Warum Enduser-Development?
- Was ist Enduser-Development?
- Herausforderung des Enduser-Developments
- Gefahren des Enduser-Developments
- Eine erfolgsversprechender Versuch: Chickenfoot
  - Konzept
  - Livedemo
- Enduser-Development in OpenSource Projekten
- Quellen
- Fragen

# Warum Enduser-Development?

Freie Universität Berlin

*Von Nöten und Talenten der Endbenutzer*

# Warum Enduser-Development?

Freie Universität Berlin

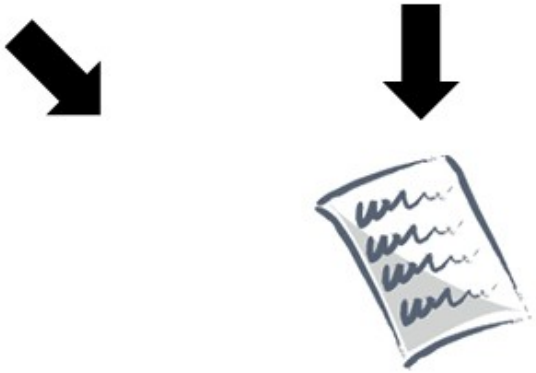
## Mafl - Hausaufgaben

Word

Formelgenerator

$$A_z = \int_0^z f_t(x)dx = \int_0^t g_t(x)dx + \int_t^z h_t(x)dx$$
$$= \int_0^t \left(-\frac{3}{t^2}x^2 + \frac{4}{t}x\right)dx + \int_t^z \frac{t^2}{x^2}dx$$
$$= \left[-\frac{x^3}{t^2} + \frac{2}{t}x^2\right]_0^t + \left[-\frac{t^2}{x^2}\right]_t^z$$
$$= -t + 2t - \frac{t^2}{z} + t = 2t - \frac{t^2}{z}.$$

Grafikprogramm



Wie wäre es statt dessen mit einem "Mafl-Word"?

# Warum Enduser-Development?



- Benutzer sind zu verschieden
  - in Bedürfnissen, Aufgaben, Endgeräten, kulturellen Hintergründen, …
    - ändern sich oft
    - sind schwer identifizierbar
  - Flexibilisierung von Geschäftsprozessen
- Entwickler fehlen die Ressourcen
  - in vielen Domänen zu wenig Wissen
  - zu wenig Zeit
  - zu wenig Motivation

# Warum Enduser-Development?



- Folge:
  - es werden mehrere *Versionen* des Programms benötigt
    - auch oft *editions* (lite, professional, enterprise, …) genannt
- oft wird Granularität zu groß
  - keine *Orange-background-blue-foreground-Edition*
- Funktionalität zu individuell oder nicht vorhersehbar
  - Spreadsheets, Plug-ins, Add-ons, Extensions, …
- "normale" Softwareentwicklung zu langsam, zu zeitintensiv, zu teuer
- exp. Wachstum von Endbenutzern im Vergleich zu Entwicklern (Boehm et al. 2000)

# Warum Enduser-Development?



- Nicht nur aus der Not:
  - Wechsel von few-to-many- zu many-to-many-Development sinnvoll
  - Komplexität der IT-Architekturen EUs zugänglich machen
    - Potenziale nutzen
  - ermöglicht “Wertzuwachs” durch Experten anderer Domänen
  - Sozio-kulturelle Aktivität
    - gewinnbringende Einflüße in die Informatik

# Was ist Enduser-Development



*Endbenutzer. Befriedigt eure eigenen Bedürfnisse!*

# Was ist Enduser-Development?

- Definitionen wie Sand am Meer
  - „Eine sozio-technische Infrastruktur um den EU seine **eigene Bedürfnisse zu befriedigen lassen** und das **effektiv** und **sicher**“ – *Rosson Pennsylvania State University*
  - “End-User Development can be defined as a set of **methods, techniques, and tools** that allow users of software systems, who are **acting as non-professional** software developers, **at some point to create, modify or extend a software artefact.**” – Liebermann, Paternò, Klann, Wulf
  - „Systeme, die Endnutzern **Möglichkeiten zur Anpassung an veränderte Anforderung** nach der eigentlichen Entwicklung bieten“ – FIT Stevens, Wulf

# Was ist Enduser-Development?

- Probleme in der Definition durch Vielfältigkeit
  - Wie viel „Entwicklung“ darf‘s denn sein?
    - Flexibel
      - verschieben der Oberfläche, …
    - Parametrisierbar
      - ausführen in verschiedenen Modi, …
    - Integrierbar (zusätzliche Komponenten einbauen)
      - Skripting, Plug-ins, …
    - Maßschneidern (frei neue Funktionen hinzuzufügen oder zu ändern)

# Was ist Enduser-Development?



- Probleme in der Definition durch Vielfältigkeit
  - 3 Dimensionen von EUD
    - Reichweite (allgemeingültig <-> domänenspezifisch)
      - Ziel ist die „eierlegende Wollmilchsau“?
      - oder Individualität in Teilbereichen (Domänen)?
    - Kommunikation (abstrakt <-> natürlich)
      - Abstrakte Sprachen und Konstrukte?
      - oder „Klickibunti“ (oder noch besser einfach vormachen)?
    - Aktivität des Benutzers (aktiv <-> passiv)
      - Gibt der Endbenutzer ausdrückliche Befehle?
      - oder geschieht ein Anpassung des Systems im Hintergrund?
- Preisfrage: Was ist EUD? Und was nicht?

# Was ist Enduser-Development?



- Beispiele für Enduser-Development:
  - flexibel / parameterisierbar
    - Vorlagen \ Templates erstellen
    - …
  - integrierbar
    - benuterfreundlich
      - spreadsheets
      - Emailfilter
      - (Makroprogrammierung)
    - mächtig
      - Scripting
      - Plug-in Programmierung
  - Maßgeschneidert
    - ???

# Was ist Enduser-Development?



- Zusammenfassend:
  - EU passen Software ihren (individuellen, wechselnden) Anforderungen an
  - EU steigen aktiv in den „Softwareerstellungsprozeß“ ein
    - erstellen, bearbeiten oder erweitern
  - „schaffen“ damit ein Softwareartefakt
  - dazu benötigen sie geeignete Schnittstellen, Architekturen, Werkzeuge…

- im Folgenden verwendete Definition:
  - EUD wird als Schöpfungsprozess von Softwareartefakten verstanden (andere Sicht aber auch legitim)
    - also: integrierbare und maßschneidernde Modifikationen

# Herausforderungen

Freie Universität Berlin

*Wie wird ein Schuh draus…?!*

# Herausforderungen



- Aufgabe: Leicht zu verstehen, leicht zu benutzen und leicht zu lernen (und zu leicht testen)
- Mächtigkeit vs. Benutzerfreundlichkeit
  - sinnvoller Kompromiss(?) finden
    - Schwierigkeiten mit der Abbildung von der menschlichen Welt in die des Computers
- Lernhürde möglichst flach halten
  - Frage: Was kann ich dem EU zutrauen?
- Motivationsfähigkeit
  - Frage: Wie stelle ich Erfolg in Aussicht?

# Herausforderungen



- Unterstützung geben
  - Frage: Was wird wie wofür gebraucht?
- Veränderbare Architekturen schaffen
  - agile Prozesse und Software erhaltende Modellarchitekturen
  - Collaborative Development
  - „design-during-use“
- Sicherheit\Qualität gewährleisten
  - Mächtigkeit vs. Sicherheit/Qualität
  - Qualitätskultur (Testing & Debuging)

# Gefahren des Enduser-Developments

Freie Universität Berlin

*Es ist nicht alle Gold was glänzt…*

# Gefahren des Enduser-Developments

## Hotelreservierung im Internet

- Wer hat das programmiert?
- Wie viel Wissen hatte der Programmierer über Sicherheit von Kreditkartentransaktionen im Internet?
- Woher hat er sein Wissen?
  - Aus einem Tutorial von 1998?
  - Aus einem Forumbeitrag von „HackerXY“?
- Wer fragt sich all diese Fragen?
  - prof. Programmierer? … hoffentlich
  - Ottonormalverbraucher (EU)? … ???

# Gefahren des Enduser-Developments

Freie Universität Berlin

- Sicherheit wird wichtiger
- EUDs sind nicht als Programmierer angestellt
  - Hilfe bei der Arbeit
  - kostengünstig (?)
- EUDs haben einen Hang zur Selbstüberschätzung
  - Gefährliches Halbwissen
    - Copy / Paste
  - "*X* für Dummies", Tutorials, HowTos
  - Trail-and-Error
    - oft keine Analyse, Design oder Tests
  - Oft ausschliesslich Kontakt zu Nicht-Professionellen
    - Fehler (oder: fehlerhafter Code) breitet sich aus

# Gefahren des Enduser-Developments

Freie Universität Berlin

"In adopting a new and complex accounting standard in a short period of time, Fannie Mae had to put in place a system and process to capture all open commitments and mark them to market …To implement this standard, Fannie Mae utilized information from its internal automated systems in conjunction with spreadsheets that made additional calculations necessary under the new rule."
**One of the spreadsheets contained a $1.2 Billion error.**
—New York Times, October 30th, 2003
*(Mary Beth Rosson - Pennsylvania State University)*

# Ein erfolgsversprechender Versuch



*Ein erfolgsversprechender Versuch*

# Ein erfolgsversprechender Versuch



- Konzept:
  - Firefox-Plugin
  - Skripte als Folge intuitiver Anweisungen
    - Befehle: enter, go, insert, pick, …
    - Elemente nach Beschriftung identifizierbar (best-fit)
  - Identifikation von Pattern
  - Aufnehmen von Aktionen und bauen von Skripten
    - Programming by example
  - Trigger für kontinuierliche Anwendung von Skripten
  - Export als Firefox-Extension

# Ein erfolgsversprechender Versuch

Freie Universität Berlin

## Chickenfoot

-Livedemo-

# EUD in OpenSource Projekten

Freie Universität Berlin

*Wie es weiter geht…*

# EUD in OpenSource Projekten



- Aufgabe: Leicht zu verstehen, leicht zu benutzen und leicht zu lernen (und zu testen)

- Mächtigkeit vs. Benutzerfreundlichkeit
- Lernhürde möglichst flach halten
- Motivationsfähigkeit
- Unterstützung geben
- Veränderbare Architekturen schaffen
- Sicherheit\Qualität gewährleisten

# EUD in OpenSource Projekten

- Welche Chance hat EUD in der OpenSource Welt?
  - Gibt es Gemeinsamkeiten, Unterstützung in den Philosophien?
  - These: OSS Entwicklung ist eine extreme Form von EUD
    - „abstrakte" Programmiersprache
    - Lernhürde sehr hoch
- Welche Eigenschaften (und Phänomene) können sich förderlich oder hemmend auswirken?
  - offener Quellcode
  - keine Lizenzgebühren
  - ausgeprägte Community
- Aber auch: Welche Möglichkeiten bringt EUD für OSS
  - Wie verändert sich z.B. das Rollenmodell von OSS

# EUD in OpenSource Projekten

- Vorgehensweise (Studienarbeit):
  - Informationen sammeln über EUD -> Seminararbeit
  - Eigenschaften, Phänomene, Bedingungen über OS-Projekte sammeln
  - Abhängigkeitsmodell (OSS-EUD) entwickeln (theoretisch)
    - *„Eigenschaft X kann sich unter Bedingung Y positiv/negativ auf die Lösung von Herausforderung Z auswirken*"
    - Thesen formulieren
      - Die für OSS typisch große Usercommunity kann die Lernhürde durch interaktive Elemente wie … verkleinern, denn …
    - In einen Modell beschreiben

# EUD in OpenSource Projekten



- Empirische Überprüfung des Modells am Projekt (praktisch)
  - OS-Projekt aus den Bereich CMS
    - Modulentwicklung
  - Untersuchung der Kommunikationsmedien, des Quellcodes, von Statistiken
    - *Konnte (auf welche Art, in welchen Zeitraum, …) einem „Endbenutzer" durch die Mailingliste das Überwinden einer Lernhürde erleichtert werden?*
  - Schlussfolgerung
    - Haben sich die theoretischen Vermutungen zu Abhängigkeiten bestätigt?
    - Sind neue Abhängigkeiten aufgetaucht?

- Ziel:
  - Fundierte Aussage treffen über OS – EUD Beziehungen
  - OS-Projekten eine Hilfe geben die sich mit EUD befassen wollen

# Quellen


Stevens, Wulf: *CoEUD – Component-based End User Development, 2006*

Fischer, Giaccardi, Ye, Sutcliffe, Mehandjiev: *Meta-Design: A Manifesto for End-User Development, 2004*

Costabile, Fogli, Letondal, Mussio, Piccinno: *Domain-Expert Users and their Needs of Software Development, 2003*

Harrison: *The Dangers of End-User Programming, 2004*

Lieberman, Paternó, Klann, Wulf: *End-User Development: An emerging paradigm, 2006*

Mørch, Stevens, Won, Klann, Dittrich, Wulf: *Component-based technologies for End-User Development, 2004*

Palanque, Bastide: *User-Centered Point of View to End-User Development, 2003*

Sutcliffe, Lee, Mehandjiev: *Contributions, Costs and Prospects for End-User Development, 2003*

Rosson: *End Users Who Meet Their Own Requirements, 2006*

Alan F. Blackwell: *End-User Developers at home, 2004*

# Zeit für Fragen

# Vielen Dank!