

- Review-Team gibt Empfehlung über die Annahme des Prüflings:

- Review-Team gibt Empfehlung über die Annahme des Prüflings:
  - akzeptieren ohne Änderungen

- Review-Team gibt Empfehlung über die Annahme des Prüflings:
  - akzeptieren ohne Änderungen
  - akzeptieren mit Änderungen (kein weiteres Review)

- Review-Team gibt Empfehlung über die Annahme des Prüflings:
  - akzeptieren ohne Änderungen
  - akzeptieren mit Änderungen (kein weiteres Review)
  - nicht akzeptieren (weiteres Review erforderlich)

- Review-Team gibt Empfehlung über die Annahme des Prüflings:
  - akzeptieren ohne Änderungen
  - akzeptieren mit Änderungen (kein weiteres Review)
  - nicht akzeptieren (weiteres Review erforderlich)
- Alle Sitzungsteilnehmer unterschreiben das Protokoll

# Checklisten: Fragenkatalog für ein Anforderungsdokument

Prüfen Sie bitte das Dokument als Vorbereitung zur Sitzung gemäß den Ihnen in Punkt D der Einladung zugeteilten Aspekten aus der folgenden Liste.

- Aspekt Form: Ist die Darstellung im Dokument sinnvoll?
  - a1 Sind Anforderungen als solche erkennbar, d.h. von Erklärungen unterscheidbar?
  - a2 Sind alle Anforderungen eindeutig referenzierbar?
  - a3 Ist die Spezifikation jeder Anforderung eindeutig?
  - a4 Sind alle Anforderungen überprüfbar formuliert?

# Checklisten: Fragenkatalog für ein Anforderungsdokument

Prüfen Sie bitte das Dokument als Vorbereitung zur Sitzung gemäß den Ihnen in Punkt D der Einladung zugeteilten Aspekten aus der folgenden Liste.

- Aspekt Form: Ist die Darstellung im Dokument sinnvoll?
  - a1 Sind Anforderungen als solche erkennbar, d.h. von Erklärungen unterscheidbar?
  - a2 Sind alle Anforderungen eindeutig referenzierbar?
  - a3 Ist die Spezifikation jeder Anforderung eindeutig?
  - a4 Sind alle Anforderungen überprüfbar formuliert?
- Aspekt Schnittstellen: Sind alle Schnittstellen eindeutig spezifiziert?
  - b1 Sind alle Objekte der Umgebung (Benutzer, andere Systeme, Basis-Software etc.) sowie alle Informationsflüsse von und nach diesen Objekten spezifiziert?
  - b2 Sind alle Benutzerklassen (Dauerbenutzer, gelegentliche Benutzer, System-Administrator, etc.) des Systems identifiziert?
  - b3 ...
  - b7 Sind Vorgaben gemacht bezüglich Verwendung von Betriebssystem-Funktionen, Bibliotheken und Hilfsprogrammen?

(Fortsetzung)

- Aspekt Vertraulichkeit: Sind die wesentlichen Aspekte des Datenschutzes berücksichtigt?
  - d1 Ist spezifiziert, welche Information vertraulich zu behandeln ist?
  - d2 Sind die Zugriffsrechte aller Benutzerklassen definiert?
  - d3 Ist definiert, gegen welche Art von unberechtigt Zugriff die Information geschützt werden muss?
- ...



- Design and Code Inspection (nach Fagan)
  - mit Einführungssitzung, Gutachter-Notizen, die abgegeben werden, Vorleser, Entscheidungskompetenz, Metriken-Erhebung.

# Varianten des Software-Reviews

- Design and Code Inspection (nach Fagan)
  - mit Einführungssitzung, Gutachter-Notizen, die abgegeben werden, Vorleser, Entscheidungskompetenz, Metriken-Erhebung.
- Schreibtischtest (besser: die Selbstkontrolle)
  - führt jeder Entwickler allein durch (vgl. Humphreys PSP);
  - ersetzt die eigentliche Prüfung nicht.

# Varianten des Software-Reviews

- Design and Code Inspection (nach Fagan)
  - mit Einführungssitzung, Gutachter-Notizen, die abgegeben werden, Vorleser, Entscheidungskompetenz, Metriken-Erhebung.
- Schreibtischtest (besser: die Selbstkontrolle)
  - führt jeder Entwickler allein durch (vgl. Humphreys PSP);
  - ersetzt die eigentliche Prüfung nicht.
- Stellungnahme
  - ist ein „off-line“-Review unter der Regie des Autors;
  - den Vorteilen (geringer Organisationsaufwand) stehen erhebliche Nachteile gegenüber (Prüfling nicht vor Weiterbearbeitung geschützt, Qualität der Prüfung und Umsetzung der Resultate nicht kontrolliert).

# Varianten des Software-Reviews

- Design and Code Inspection (nach Fagan)
  - mit Einführungssitzung, Gutachter-Notizen, die abgegeben werden, Vorleser, Entscheidungskompetenz, Metriken-Erhebung.
- Schreibtischtest (besser: die Selbstkontrolle)
  - führt jeder Entwickler allein durch (vgl. Humphreys PSP);
  - ersetzt die eigentliche Prüfung nicht.
- Stellungnahme
  - ist ein „off-line“-Review unter der Regie des Autors;
  - den Vorteilen (geringer Organisationsaufwand) stehen erhebliche Nachteile gegenüber (Prüfling nicht vor Weiterbearbeitung geschützt, Qualität der Prüfung und Umsetzung der Resultate nicht kontrolliert).
- Structured Walkthrough
  - ist die Billig-Variante des Reviews: Autor ist Moderator;
  - er kompensiert durch seine Präsentation die Einsparung (oder Reduktion) der Vorbereitung;
  - während seiner Vorbereitung entdeckt er selbst viele Fehler.

- Perspektivisches Review
  - Prinzip:
    - Prüfling wird von Gutachtern mit unterschiedlichem Interesse geprüft
    - Gutachter haben konkreten Auftrag: Prüfling (hypothetisch) benutzen
  - Beispiel:
    - Tester → Testfälle entwickeln
    - Benutzer → System benutzen
    - Handbuchautor → Handbuch schreiben
    - Architekt → Entwurf erstellen

Empirische Studien legen nahe (Berling und Runeson 2003; Fusaro u. a. 1997; Hayes 1999; Miller u. a. 1998; Porter u. a. 1995; Porter und Votta 1998; Sandahl u. a. 1998):

- Reviews, die auf Checklisten basieren, sind effektiver als Ad-Hoc-Reviews;
- perspektivische Reviews sind effektiver und effizienter als Reviews, die auf Checklisten basieren:
  - mehr Fehler/Mängel gefunden
  - mehr Fehler/Mängel in gleicher Zeiteinheit gefunden

# Woran scheitern Reviews? Fallen und Gegenmittel

Probleme bei der Ein-/Durchführung von Reviews:

- Es fehlen gute Moderatoren
  - Leute aussuchen und ausbilden

# Woran scheitern Reviews? Fallen und Gegenmittel

Probleme bei der Ein-/Durchführung von Reviews:

- Es fehlen gute Moderatoren
  - Leute aussuchen und ausbilden
- Es fehlen Bezugsdokumente
  - suchen, anpassen, bereitstellen



# Woran scheitern Reviews? Fallen und Gegenmittel

## Probleme bei der Ein-/Durchführung von Reviews:

- Es fehlen gute Moderatoren
  - Leute aussuchen und ausbilden
- Es fehlen Bezugsdokumente
  - suchen, anpassen, bereitstellen
- Entwickler haben Angst
  - erste Reviews gründlich vorbereiten (und auf die Ängste eingehen, selbst wenn sie geleugnet werden)

# Woran scheitern Reviews? Fallen und Gegenmittel

## Probleme bei der Ein-/Durchführung von Reviews:

- Es fehlen gute Moderatoren
  - Leute aussuchen und ausbilden
- Es fehlen Bezugsdokumente
  - suchen, anpassen, bereitstellen
- Entwickler haben Angst
  - erste Reviews gründlich vorbereiten (und auf die Ängste eingehen, selbst wenn sie geleugnet werden)
- Reviews beißen sich an Äußerlichkeiten fest
  - Bedeutung der Äußerlichkeiten klarstellen, aber dann weitergehen;
  - beim zweiten Review sicherstellen, dass die Äußerlichkeiten in Ordnung sind.

# Woran scheitern Reviews? Fallen und Gegenmittel

## Probleme bei der Ein-/Durchführung von Reviews:

- Es fehlen gute Moderatoren
  - Leute aussuchen und ausbilden
- Es fehlen Bezugsdokumente
  - suchen, anpassen, bereitstellen
- Entwickler haben Angst
  - erste Reviews gründlich vorbereiten (und auf die Ängste eingehen, selbst wenn sie geleugnet werden)
- Reviews beißen sich an Äußerlichkeiten fest
  - Bedeutung der Äußerlichkeiten klarstellen, aber dann weitergehen;
  - beim zweiten Review sicherstellen, dass die Äußerlichkeiten in Ordnung sind.
- Bezugsdokumente sind ungeeignet
  - diskutieren und verbessern

- Zeitdruck sabotiert Prüfungen
  - klare Entscheidungen über die Prioritäten treffen

# Woran scheitern Reviews? Fallen und Gegenmittel II

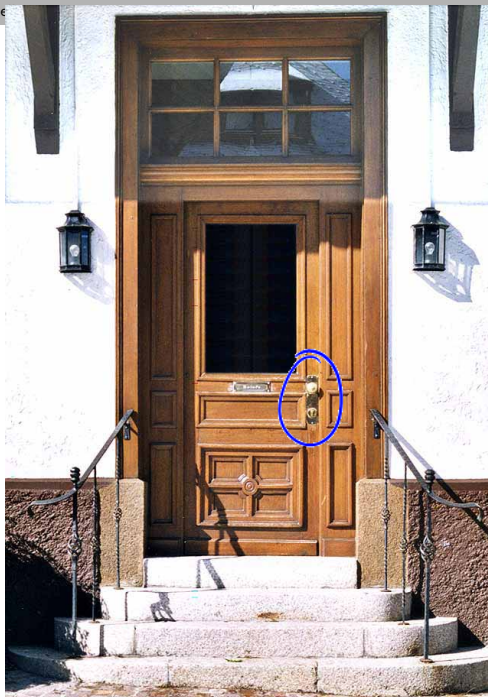
- Zeitdruck sabotiert Prüfungen
  - klare Entscheidungen über die Prioritäten treffen
- Geprüfte Dokumente werden verändert
  - Konfigurationsmanagement verbessern

# Woran scheitern Reviews? Fallen und Gegenmittel II

- Zeitdruck sabotiert Prüfungen
  - klare Entscheidungen über die Prioritäten treffen
- Geprüfte Dokumente werden verändert
  - Konfigurationsmanagement verbessern
- Interesse an Reviews kühlt sich ab („Jetzt ist doch eigentlich alles in Ordnung!“)
  - Statistiken führen, Erfolg nachweisen

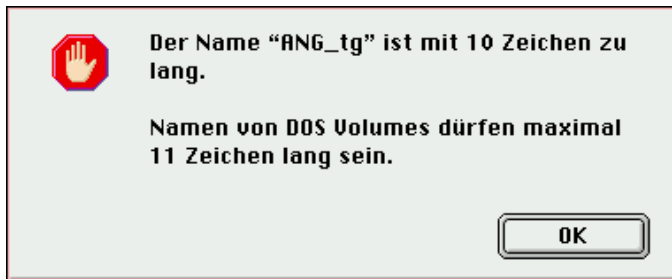
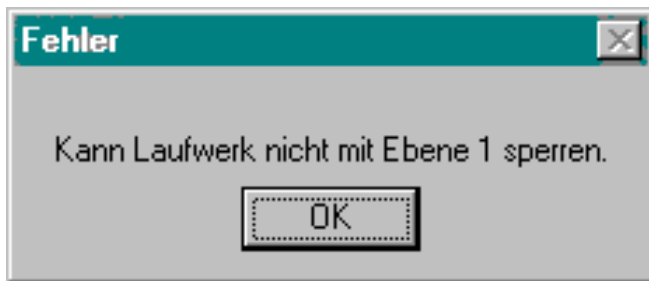
## 1 Entwurf von Benutzungsschnittstellen

- Motivation
- Lernziele
- Ergonomie
- Psychologische und kognitive Grundlagen
- CLI versus GUI
- Geschichte graphischer Benutzungsschnittstellen
- Interaktionsmechanismen
- Software-ergonomische Richtlinien
- Usability messen und verbessern
- Applehouse
- Zusammenfassung





# Negativbeispiele der Softwaretechnik



Ergonomische Software erstellen können:

- Bedienelemente graphischer Benutzungsschnittstellen kennen und einsetzen können
- Ergonomie bewerten können

## Definition

**Ergonomie:** (von ergon (Arbeit, Werk) und nomos (Gesetz, Regel)) ist die Wissenschaft der Arbeitsbedingungen und deren optimale Anpassung an den Anwender.

## Definition

**Ergonomie:** (von ergon (Arbeit, Werk) und nomos (Gesetz, Regel)) ist die Wissenschaft der Arbeitsbedingungen und deren optimale Anpassung an den Anwender.

- “Design for Use”
- Anpassung von Maschinen, Computern und Systemen an menschliche (Denk- und Kommunikations-) Fähigkeiten und Bedürfnisse
- zentral ist Schnittstelle zwischen Mensch und Maschine (Mensch-Computer-Interaktion)
- Auswirkungen auf zum Beispiel: Arbeitsabläufe, Menü-Hierarchien, Kommandosprachen, Farb- und Schriftwahl und die Funktionsaufteilung zwischen Mensch und Computer

## Definition

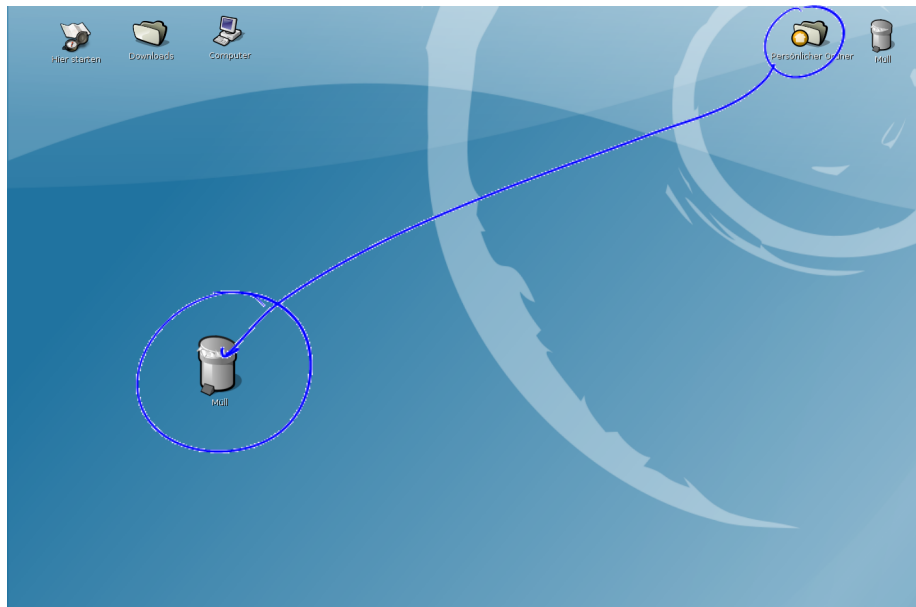
**Usability:** *eines Produktes* ist das Ausmaß, in dem es von einem *bestimmten Benutzer* verwendet werden kann, um *bestimmte Ziele* in einem *bestimmten Kontext* effektiv, effizient und zufriedenstellend zu erreichen. — Part 11: “Guidance on Usability” (ISO 9241-11:1998 1998)

- übersetzt als „Benutzerfreundlichkeit“ oder „Benutzbarkeit“
- Verbesserung ist Ziel der (Software-)Ergonomie



- Kurzzeitgedächtnis
  - Information strukturieren und begrenzen:  $7 \pm 2$
- menschliche Gestaltwahrnehmung
  - bei Präsentation beachten
    - z.B. Superzeichenbildung: 0 4 2 1 | 2 1 8 | 2 4 2 1 versus 0421/218-2421
- geteilte Aufmerksamkeit
  - Fortsetzung nach Unterbrechung unterstützen
- begrenzte Konzentrationsfähigkeit
  - nicht überfordern, Sicherheiten einbauen
- Unerfahrenheit verunsichert
  - Metaphern verwenden, z.B. Ikonen wie Mülleimer, Briefkasten
  - explorative Untersuchung unterstützen

# Metaphern



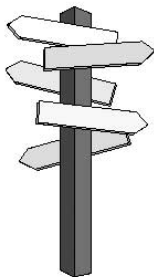
# Metaphern





# Vergleich von *GUI* und *Command Line Interface*

<b>Command Line Interface (CLI)</b>	<b>Graphical User Interface (GUI)</b>
komplexe Operationen	einfache Operationen
intensive Nutzung, Experten	gelegentlicher Einsatz, leichte Erlernbarkeit
funktionale Bedienung, Berechnungen	Objektorientierung,

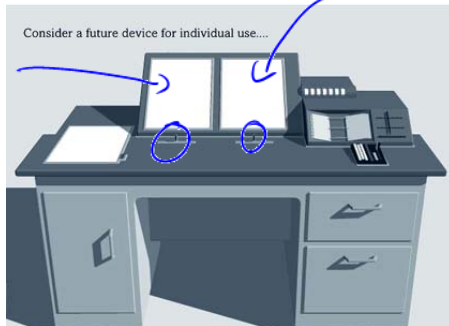


## Meilensteine in der Geschichte graphischer Benutzungsschnittstellen

# Die ersten Ideen

## Memex (**M**emory **E**xtender) von Bush (1945)

- Kompakt-Analog-Rechner zur Verwaltung verlinkter Informationen
- berührungssensitive Bildschirme projizieren Mikrofilme
- Mikrofilme verknüpft
- Vor- und Zurückblättern über Hebel



# Semi Automatic Ground Environment

- Bedienung durch Laien
- Lichtgriffel (Light-Pen)



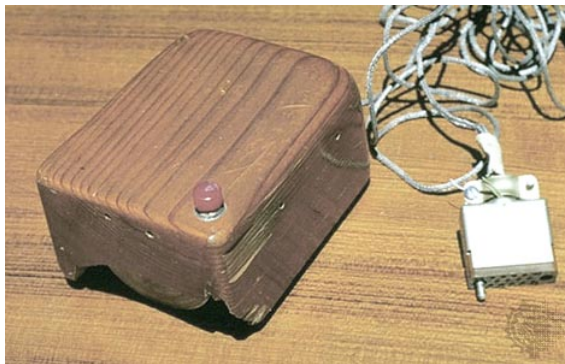
# Sketchpad

- von Ivan Sutherland (1963)
- Objektorientierung statt Bitmaps
- Generische Operationen für verschiedene Objekte
- Constraints zu den graphischen Objekten (z.B. Länge)
- Copy & Paste



# Computermaus

- von Douglas Engelbart und Bill English (1963)
- relative Positionsveränderung



- Frühe Ideen: *Xerox Palo Alto Research Center* (1970)
- Prinzipien:
  - abstrakte Datensicht, Visualisierung
  - Raum und Objekte
  - Metaphorisierung: Schreibtisch (Desktop)
  - “What you see is what you get” (Texteditor *BRAVO*, Vorläufer von *Microsoft Word*)
  - einheitliche Visualisierung und Interaktion
  - wenige Modi, generische Kommandos

# Xerox Star GUI

