

# *Präsentationsthemen:*

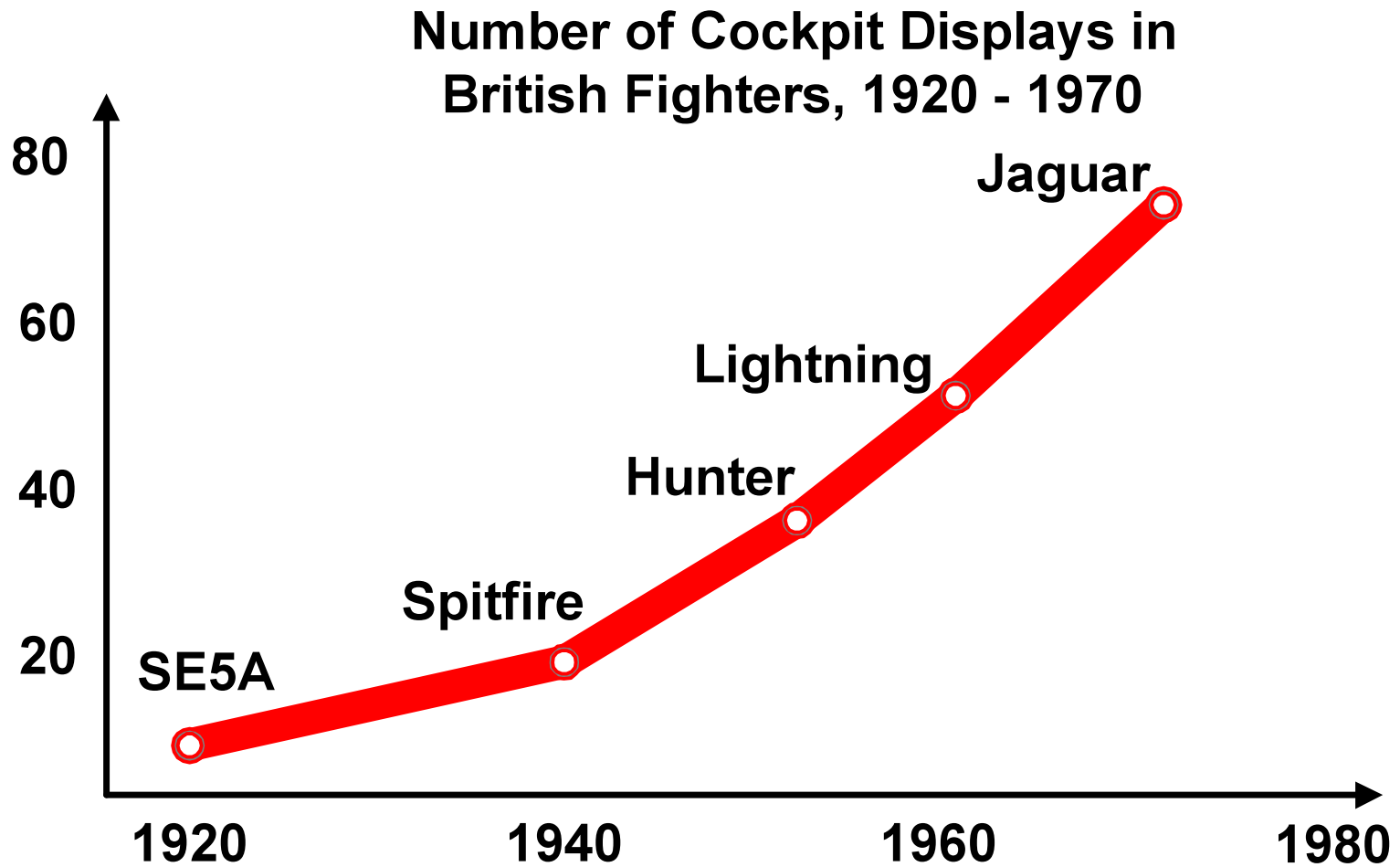
- ① Hintergründe
- ② Analogie in Cockpits und Desktops
- ③ Kostenfolgen von unbrauchbaren User Ifs
- ④ Design-Prinzipien
- ⑤ Konsequenzen für Web Interfaces
- ⑥ Allgemeine Eindrücke (mündlich)

# 1. Hintergründe



- ★ 50 jähriges Jubiläum des ACM
- ★ Grösster Amerikanischer Informatiker-Verband
- ★ CHI - Computer Human Interaction = grösster Teilverband
- ★ Konferenz-Thema: **Looking to the Future**
- ★ **Strategische** Ansätze in der Software-Entwicklung

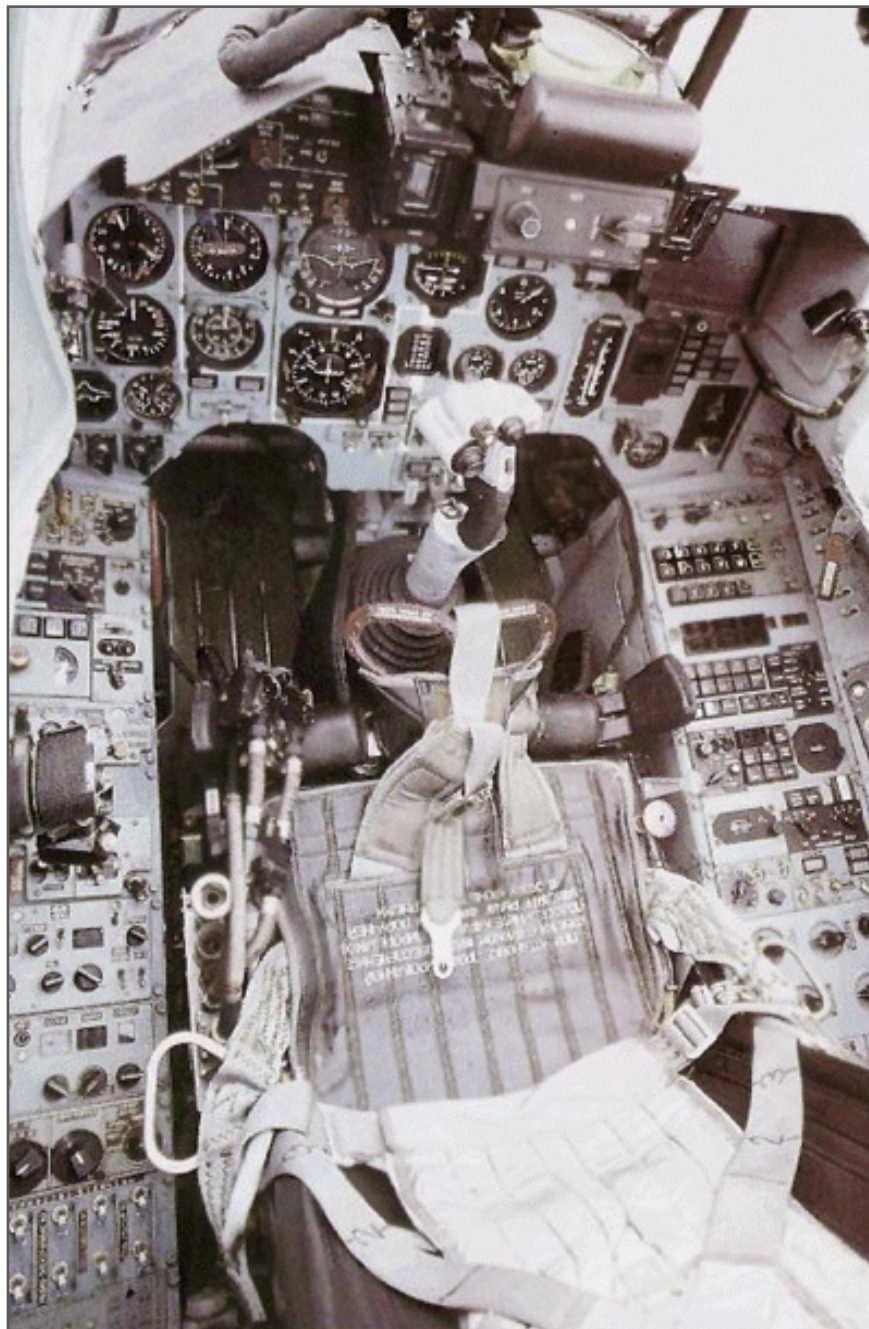
## 2. Analogie in Cockpits und Desktops



Human Factors Engineering began when "pilot error" accidents followed the growth of cockpit technology in WWII





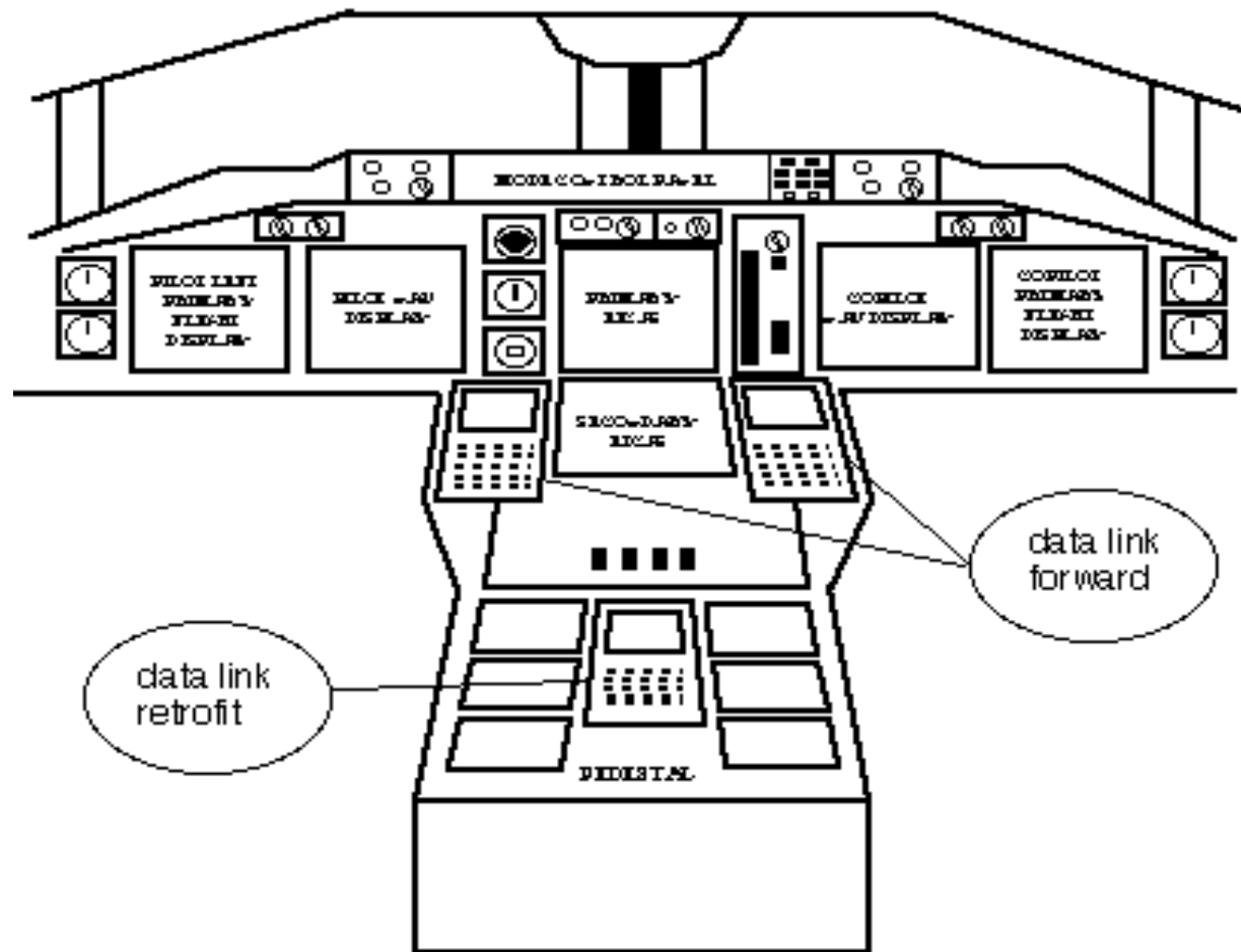


## 2.3 *Flugzeug-Cockpits*

Heutige Flight-Desks verwenden bewährte Design-Prinzipien in Cockpit-Computern für:

- Grössere Sicherheit
- Grösseren Komfort
- Hohe Wirtschaftlichkeit
- Eine Reduktion der Arbeitsbelastung

Figure 1. Boeing 747-400 cockpit layout



## 2.4 Ähnliche Methoden

Generelle Ähnlichkeiten von Design-Methoden für Cockpits und Desktops:

- ⇒ Piloten/Benutzer arbeiten mit beim Design
- ⇒ Konstruktion von Prototypen zum Testen von Konzepten und zur Problembereinigung vor der Auslieferung
- ⇒ Piloten/Benutzer testen Prototypen mit repräsentativen Aufgaben
- ⇒ Akzeptanz verlangt messbare Kriterien



## 2.5 Motivation zum Desktop Design

Höhere Benutzer-Produktivität beeinflusst meistens gleichzeitig die Applikationen:

- ⇒ **USABILITY** ist die Definition des Endbenutzers für System-Qualität
- ⇒ **USABILITY** ist der Punkt, wo wir eine Verbindung oder einen Graben erzeugen zwischen Computereinsatz und Produktivitätserhöhung
- ⇒ **USABILITY**-Messungen sind oft der Ansatz zur Verbesserung der Geschäftsprozesse

### ***3. Kostenfolgen von unbrauchbaren User IFs***

Zuviele Systeme sind nur technisch brillant und erfüllen die Erwartungen nicht bezüglich Gewinn und ROI !

Die Benutzer finden solche Systeme

- ➡ verwirrend statt begreifbar
- ➡ nicht folgerichtig
- ➡ extrem lernaufwendig, viele Umdenkprozesse
- ➡ fehleranfällig
- ➡ nicht ermutigend zum Entdecken

## ***3.1 Kosten von Un-Usability bei Entwicklern***

**Das Benutzer-Interface erfordert ca. 50% des  
Entwicklungs-Aufwandes einer Applikation**

**Das Fehlen von systematischen Anforderungen an das Design**

- ↘ führt zu Fehlstarts
- ↘ erhöht den Entwicklungsaufwand
- ↘ Verschlingt Unterhaltsbudget mit "Usability Patches"
- ↘ führt zu komplexerem Code
- ↘ senkt die Befriedigung der Kunden

## 3.2 Kostenfolgen bei den Benutzern

### Messbare Usability-Ziele von Butler, et al

	Geplante Werte	Akzeptierte Maximal Werte	Reale Werte
Produktivität	10 min/Record	15 min/Record	30 min/Record
Lernaufwand	90 min	120 min	2 Wochen

## 4. Design-Prinzipien









Wie wird das Arbeitsziel des Benutzers optimal unterstützt?

### Top 10 Mistakes in Web-Design

- ☞ Frames einsetzen
- ☞ Zu früher Einsatz von neuen Technologien
- ☞ Zu lange Seiten (90% der Benutzer scrollen nicht!)
- ☞ Komplexe URL's
- ☞ Fehlender Navigation-Support, Orphan Pages
- ☞ Unstrukturierte Seiten (**markieren!**)
- ☞ Verwendung von Nicht-Standard-Farben für Links
- ☞ Veraltete Informationen
- ☞ Lange Ladezeiten, max. 15 Sekunden
- ☞ Permanent bewegte Bilder



## 4.1 Positive Merkmale

-  Speziell aufbereitete Texte
-  Orientierungshilfen
-  Einfache, kleine Buttons mit alt-Text
-  Horizontale Gliederung ist besser als vertikale
-  Kleine (Anzahl kB) Bilder, Netscape Farben
-  Graphische Hinweise
-  Trennen von Inhalt und Navigation
-  Aktionsmöglichkeiten verwenden

## ***5. Konsequenzen für Web Interfaces***

- ➔ Einfach
- ➔ "Gags" nur wenn Ziel-unterstützend
- ➔ Inhalt
- ➔ Funktionalität
- ➔ Alternativen
- ➔ User tested