



















Allen EBÜS-Komponenten liegen folgende Design-Prinzipien zu Grunde:

- **Form follows function** – die Bedienoberfläche soll der Funktion dienen und den Anwender führen. Im Vordergrund steht, die **Arbeitsabläufe der Anwender** möglichst gut zu unterstützen.
- Deshalb keine aufwändige Grafik und ungewöhnliche Bedienelemente, sondern sachorientierte platzsparende übersichtliche Dialoge auf Basis der bei Windows **üblichen Bedienelemente**, Fensterstrukturen, Tastenkombinationen und Bedienabläufe.
- **Einheitliches Erscheinungsbild** und **gleichartige Bedienung** ähnlicher Funktionen (=> Wiederverwendung von Konfigurationsdateien und Dialogen)
- Es wird eine möglichst **flache Fensterhierarchie** benutzt, weil tief verschachtelte Strukturen und viele Fenster übereinander für die meisten Anwender sehr unübersichtlich sind.
- Die Anordnung der Bedienelemente unterstützt die Benutzerführung, indem sie den Arbeitsabläufen entspricht: **Systematisches Abarbeiten** von oben nach unten und von links nach rechts.
- Für **kurze Mauswege** und um den **verfügbaren Platz optimal** zu **nutzen**, lassen wir zwischen den einzelnen Bedienelementen (Buttons, Listen, Edit-Felder etc.) nur gerade so viel Zwischenraum, wie für eine **übersichtliche Gliederung** nötig ist. EBÜS besteht aus zahlreichen Komponenten, die ggf. nebeneinander auf dem Bildschirm Platz finden sollen, und an manchem Server im Geräteschrank steht nur ein kleiner Monitor zur Verfügung: Da zählt jeder gesparte Pixel, damit der Anwender alle wichtigen Informationen und Bedienelemente gleichzeitig im Blick hat.
- Zusammengehörige Bedienelemente (etwa Schaltflächen oder Textfelder, die zu einer Funktionsgruppe gehören), werden **ohne Zwischenraum** direkt aneinander gereiht.
- **Feste Positionen** aller Bedienelemente innerhalb des Rahmenfensters
- Die Bedienelemente werden soweit möglich exakt an gemeinsamen **Fluchtlinien** ausgerichtet.
- Bei horizontal nebeneinander angeordneten Bedienelementen (z.B. Texteingabefeld mit zugehörigem Label) wird darauf geachtet, dass die **Basislinie der Schrift exakt in gleicher Höhe** liegt.
- Bedienelemente werden durch **ToolTips** erläutert (Mauszeiger verweilt → Hinweistext anzeigen).
- Möglichst wenig Schritte bis zum Ziel: Alle überflüssigen Mausklicks und Tastendrücke einsparen.
- Häufig benötigte Funktionen sind auch über Tastatur („**Shortcuts**“) erreichbar.
- **TabStops** erleichtern und beschleunigen die Eingabe über die Tastatur, indem mit der TAB-Taste in sinnvoller Reihenfolge genau nur die relevanten Bedienelemente angesprungen werden.
- Die wichtigsten Funktionen auf **1 Mausklick** direkt im Zugriff.
- Idealbild: **mechanisches Schaltpult** → Druck auf einen Taster löst Funktion aus, Reaktion darauf ist unmittelbar erkennbar.
- Alle wichtigen Informationen **auf einen Blick**. Videofenster nicht verdecken, denn sie könnten jederzeit wichtige Vorkommnisse zeigen.
- Keine optischen Schnörkel und Verzierungen, sondern volle Konzentration auf die nach Wichtigkeit abgestuft hervorgehobenen Sachinformationen.
- Intuitive und semantisch **einheitliche Farben**:

Farbe		Bedeutung	Englisch	Bildtitelzeile	Schalter
Grau		inaktiv	idle	Synchronbild	unbekannt
Gelb		aktiv, beschäftigt	active, busy	Voralarm	an, geschlossen
Grün		bereit, in Ordnung	ready, OK	Live	aus, offen
Cyan		warte...	waiting...	Historie	wird geschaltet
Blau		Achtung!	attention!	Referenzbild	
Magenta		Warnung	warning	Differenzbild	
Rot		Fehler	error	Alarm	Fehler
Weiß		Nichts	nothing	kein Bild	
Schwarz		aus	off		

- Die Farbe Rot ist für wirklich wichtige Dinge reserviert:
  - Alarmer, die bearbeitet werden müssen
  - Fehler, auf die reagiert werden muss
- Jedes Rahmenfenster ist - soweit sinnvoll und möglich - „**resizeable**“ (vom Anwender in der Größe veränderbar), damit es optimal an die verfügbare Bildschirmfläche angepasst werden kann.

- Bei Vergrößerung / Verkleinerung von Windows-Fenstern (resize) werden Listen und mehrzeilige Text-Felder in ihrer Größe an den verfügbaren Platz angepasst, damit sie **möglichst viele Informationen auf einen Blick** anzeigen, ohne dass der Anwender scrollen oder blättern muss.
- Die **Schriftgröße** in Listen und mehrzeiligen Text-Feldern kann zentral eingestellt werden, um ein bequemes Ablesen auch aus größeren Entfernungen zu ermöglichen.
- **Klare Benutzerführung**: Anwender erhält wo nötig Hinweise, was als nächstes zu tun ist.
- Wichtige Schaltflächen werden mit aussagekräftigen Symbolen (**Icons**) ergänzt, damit der Anwender sie schneller findet und intuitiv zuordnen kann.
- **Fehlbedienungen** wie etwa unzulässige Werte werden direkt bei der Eingabe abgefangen und durch sinnvolle Hinweise in der Nähe beantwortet.
- Auf einen bevorstehenden **Lizenzablauf** wird rechtzeitig deutlich wiederholt aufmerksam gemacht.
- **Direktes Feedback**: Anwender sieht sofort Reaktion auf seine Aktion, damit er nicht ungeduldig wird. Wartezeiten werden mit **Fortschrittsanzeige** oder **Statustexten** überbrückt, damit der Anwender sieht, dass die Anwendung noch „lebt“ und nicht aus Ungeduld wild herumklickt.
- Die Fortschrittsanzeige sollte dem Anwender möglichst konkret zeigen, wie lange es noch dauert.
- Blockierende Meldungen („Message-Boxen“) möglichst vermeiden, denn sie halten Bedienabläufe extrem auf, insbesondere wenn sie gehäuft auftreten. Stattdessen empfiehlt sich eine einheitliche **zentral konfigurierbare Meldungsbearbeitung** und -anzeige, bei der beispielsweise bestimmte Meldungstypen automatisch per E-Mail an den Systembetreuer weitergeleitet werden können, damit der zeitnah reagieren kann.
- Mehrfachmeldungen des gleichen Inhalts unterdrücken, mehrere Meldungen des gleichen Typs zusammenfassen.
- Meldungen automatisch zurückziehen, falls ihre Ursache abgestellt wurde; Klarmeldung protokollieren.
- Für eine systematische und **einheitliche Meldungsbearbeitung** und -anzeige werden folgende Meldungsebenen verwendet:

Index	Messagelevel	Symbol	Beschreibung
0	Obsolete		Klarmeldung: Meldung hat sich erledigt und wird zurückgezogen
1	Info		Nur zur Information; keine Aktion erforderlich
2	Question		Frage, die vom Anwender beantwortet werden muss
3	Alarm		Alarmmeldung, die bearbeitet werden muss
4	Warning		Warnhinweis bitte nachgehen, um Folgefehler zu vermeiden
5	External failure		Störungsmeldung von einer externen Komponente
6	Invalid Config		Unzulässige Konfiguration, Systembetreuer benachrichtigen
7	Not enabled		Funktion ist für diesen Anwender nicht freigegeben
8	Error		Programmfehler, bitte Hinweise beachten
9	Fatal		Fehler, der eine weitere Programmausführung vereitelt

- Störungsmeldungen, Hinweise auf unzulässige Konfigurationen und nicht freigegebene Funktionen sind klar von Programmfehlern zu unterscheiden und entsprechend darzustellen, damit die eigentliche Quelle des Problems schnell erkannt und beseitigt werden kann.
- Die Meldungen sollen die Ursache des Problems möglichst präzise, aber auch verständlich beschreiben und gezielte Hinweise enthalten, wie das Problem behoben werden kann.
- Der Zustand der verschiedenen Systemkomponenten wird übersichtlich mit farbigen **Status-LEDs** angezeigt. Hier sollen auch alle sich möglicherweise wiederholenden oder permanenten Störungen signalisiert werden, damit nicht zu viele Einzelmeldungen auflaufen. Bei Anklicken einer der LEDs wird ein Fenster mit Detail-Informationen angezeigt.



Dipl.-Ing. Hardo Naumann, 12.12.2017