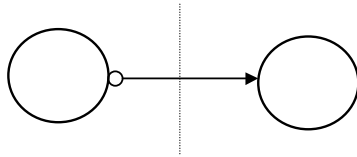


1 Kopplung = Coupling

Es geht um die Daten die zwischen den Unterprogrammen fliesen.



Datenkopplung (Data Coupling) geht nur dann gut, wenn wenig Daten übergeben werden.

- Pointer
- + es wird „wenig“ übergeben, da ein Pointer nur ein String ist
 - es wird damit aber zB.: das ganze Recordset übergeben
 - geht nicht wenn UP auf verschiedenen Rechnern laufen

Wenn Daten übergeben werden, sollte nach Möglichkeit ein Pointer übergeben werden.

zur Erinnerung:	call by value	=> Echtdaten werden übergeben
	call by reference	=> Pointer wird übergeben

1.1 Globale Kopplung = Global Coupling

!! ist sehr schlecht = unbedingt vermeiden !!

Es ist schwer zu debuggen, da man nicht weiß, wer wo was hinschreibt. Außerdem muß man, wenn man die Struktur einer Variablen ändert, 15 Programmierer verständigen.

Bei IVAN ist die größte Variable die Datenbank selbst, da jeder darauf zugreifen und sie löschen und speichern kann.

1.2 Inhaltskopplung = Content Coupling

Man spricht von Inhaltskopplung, wenn ein UP den Code eines anderen UP's ändert.

2 Gestaltung von Benutzeroberflächen

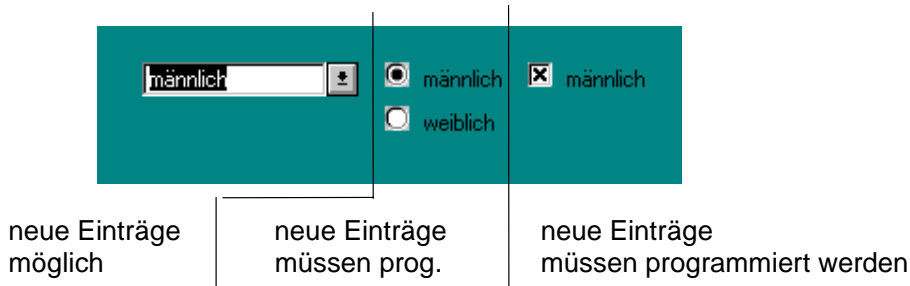
2.1 Farben

- augenfreundlich
- pro Seite nicht mehr als zwei Farben
- große Flächen in dezenteren Farben => kleine können grell sein, aber nicht mehr als zwei verschieden grelle Farben verwenden

2.2 Layout

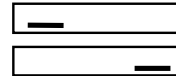
- Wenn von Papier auf EDV umgestellt wird, darauf achten, daß die Eingabemaske dem Papiervorgänger ähnlich sieht, da die EDV dann eher angenommen wird.
- Formulare sollten einander ähnlich im Aufbau sein (Style Guide).

- Entfernung des Benutzers vom Bildschirm muß beachtet werden => dementsprechende Schriftgröße (Benutzer rennt hin und her => große Schrift => wenig auf einer Maske)
- Zusammengehöriges zusammen (Vor- und Nachname)
- am Kulturkreis des Benutzers orientieren (Arabien => schreiben von rechts nach links)
- zwischen Lesefeldern, Lese-Kann-Schreibfeldern, Lese-Muß-Schreibfeldern unterscheiden. Wenn dies von der Anwendung abhängig ist, zur Laufzeit ändern.
- Die richtigen Steuerelemente wählen:



mittlere Lösung ist am Besten (wenn aber zB.: steuerpflichtig dazukommt, ist die letzte Lösung am Besten, da man alles extra ankreuzen kann)

- Beim Verlassen keine Datenbankabfragen (Datenbankaktionen) mehr starten; zB.: Datumsüberprüfung auf OK-Button verlegen und nicht beim Schließen
- Felder mit links-rechts-Auslegung nicht untereinander



- Felder nicht eingegeben werden können, inaktiv setzen.
- Statuszeilentext ist wünschenswert (Suchmodus, Lesemodus, Änderungsmodus, ...)
- selbstsprechende Fehlermeldungen (nicht: Falsche Eingabe, sondern Warum)
- bei riskanten Sachen: Rückfragen (zB.: Löschen)
- Hot-Keys von Vorteil, da Profis kaum die Maus benutzen
- Eingabe am Benutzer orientieren

55 jährige können keine Mäuse benutzen

Kinder können mit der Maus besser als mit der Tastatur umgehen

bei Behinderten: wie können diese Eingeben

bei Börsenmarklern: keine piepsende Stimme

gleiches gleich benennen: Nach- + Zuname = falsch

- Schriften wie Farben

3 Rapid Prototyping

- Als erstes einen Prototyp - bei dem noch nichts funktioniert - an den Kunden, damit dieser die Oberfläche sehen kann, und sich mit ihr anfreundet. Dies sollte ein paar Tage nach der Erstellung des Pflichtenheftes geschehen.
- Kunde kann dann bevor die Programmierung beginnt, sagen ob er mehr oder weniger Steuerelemente, andere Farben, ... haben will. Der Vorteil darin ist, daß es beim Endtermin zB.: 3-4 Jahre später kein bödes erwachen gibt.
- es handelt sich um einen Teil der Spezifikation

4 Projektplanung (Softwareprojektplanung)

vor der Spezifikationsphase

es muß entschieden ewrden, ob: Case Tool,)
 Prototyping, } ja/nein
 Objektorientiert,)
 welche Phasen wie lange,

...

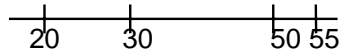
Als erstes müssen die ZIELE festgelegt werden => daraus ergeben sich dann die Aufgaben, die wiederum in Teilgebiete unterteilt werden müssen.

Für den Projektleiter ist wichtig: WIEVIEL wird es kosten,
 WIELANGE wird es dauern,
 WER macht WAS?

Es gibt verschieden Methoden, die Kosten eines Projektes abzuschätzen. Die Funktion-Point-Methode dient nur der Abschätzung der Programmierkosten.

Um alle Aufwände, und vorallem die Dauer, zu ermitteln, dient die BREITBAND DELPHI METHODE:

Verschieden Experten setzen sich zusammen und reden über das Projekt, dann tragen sie geheim (jeder für sich) auf einer Linie die von ihnen geschätzten Mann-Monate ein. Wenn alle Zettel abgegeben worden sind, werden alle Ergebnisse auf einem Diagramm zusammengefaßt, die Experten wissen jetzt was die anderen meinen, aber nicht von wem welche Abschätzung stammt.



Das ganze beginnt jetzt von vorne nochmals, bis man zu einem gemeinsamen Ergebnis kommt.

Der Vorteil dieser Methode liegt darin, daß nicht ein Mittelwert, sondern ein Wert der allgemeinen Zustimmung genommen wird, und meist sehr gute Ergebnisse erzielt werden. Kleine Fehler können aber zu großen Verzögerungen führen.

Hierzu ist noch zu sagen, daß die Arbeitsumgebung wesentlichen Einfluß auf das Projekt hat. Kleinere Räume mit viel Tageslicht und Pflanzen erhöhen den Eifer. Personen die nicht „miteinander können“, nie in einen Raum setzten, da sie nie zu einem gegenseitigen „Mutmachen“, sondern eher zum Mißmachen neigen.