

INHALTSVERZEICHNIS

Fragenkatalog

Prüfungsfragen

1 Mensch-Computer-Interaktion Fragenkatalog

HINWEISE: keine Gewähr für Richtigkeit der hier angegebenen Lösungen. Bitte unbedingt das Buch *Human-Computer Interaction* von Alan Dix als Einstieg lesen.

1.1 Grundlagen

1.1.1 Was ist HCI und was untersucht es?

HCI ... Human-Computer Interaction

HCI untersucht:

- Kontext von Computern (Arbeitsplatz, Auto, Haus, Öffentlicher Platz, ...)
- Fähigkeiten des Menschen
 - menschliche Informationsverarbeitung (visual overload, Motorik, Kinder/Behinderte/Alte)
 - Kommunikation (sozial)
 - Ergonomie (individuelles Vermögen und Grenzen)
- Entwicklungsprozess (Evaluation, Probleme der Nutzer mit Software, Nutzersicht/Expertensicht)
- Architektur der Schnittstellen
 - GUI (Maus-basiert, Desktop, Web-basiert)
 - Multimediale UI (Games, Broadcasting, Mobile, Ubiquitous)
 - Multimodale UI

1.1.2 Was ist MCI?

MCI ... Mensch-Computer-Interaktion

Interaktion eines oder mehrerer Menschen mit einem oder mehreren Computer. Bezeichnet auch Kommunikation von Menschen mittels eines Computers.

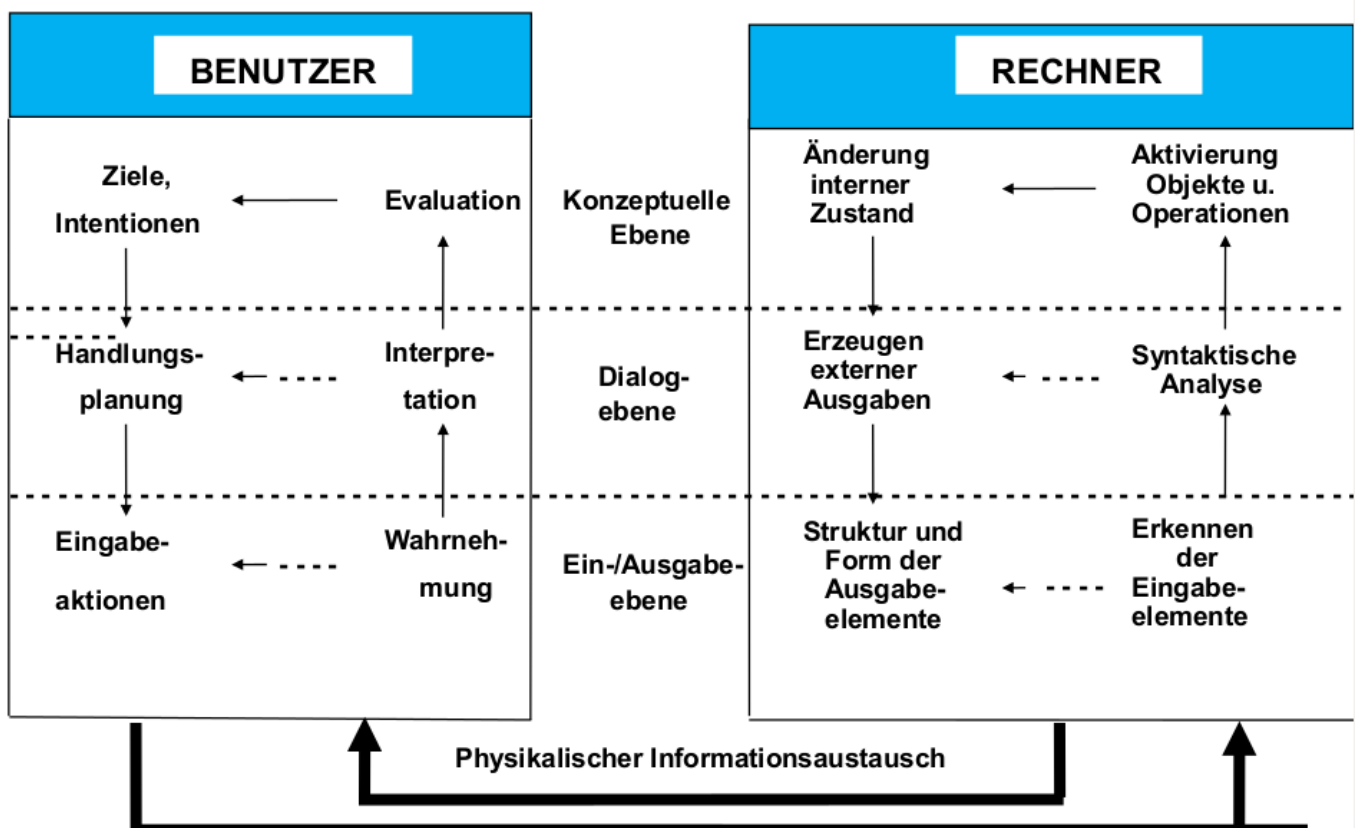
1.1.3 Welche Paradigmen der Interaktion kennen Sie?

- Time Sharing
- Video Display Units
- Programmier-Toolkits
- Personal Computing
- Fenstersysteme und WIMP
- Metaphern
- Direkte Manipulation
- Sprache vs. Aktionen
- Hypertext
- Multimodalität
- Computer Supported Cooperative Work
- World Wide Web
- Agenten-basierte Benutzungsoberfläche
- Ubiquitäre Systeme (allgegenwärtig)
- sensorbasierte und kontextabhängige Interaktion

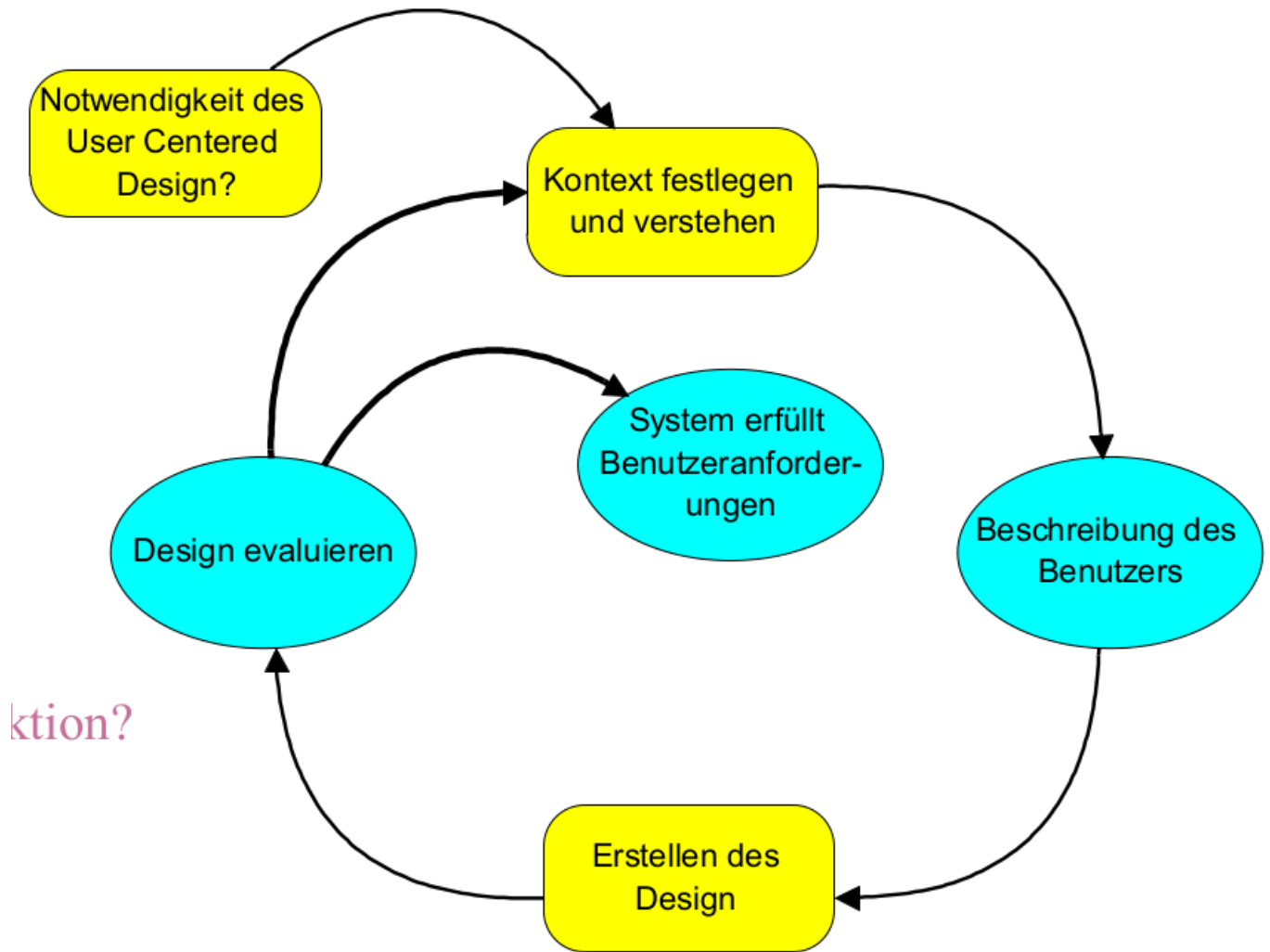
1.1.4 Erklären Sie das Schichtenmodell.

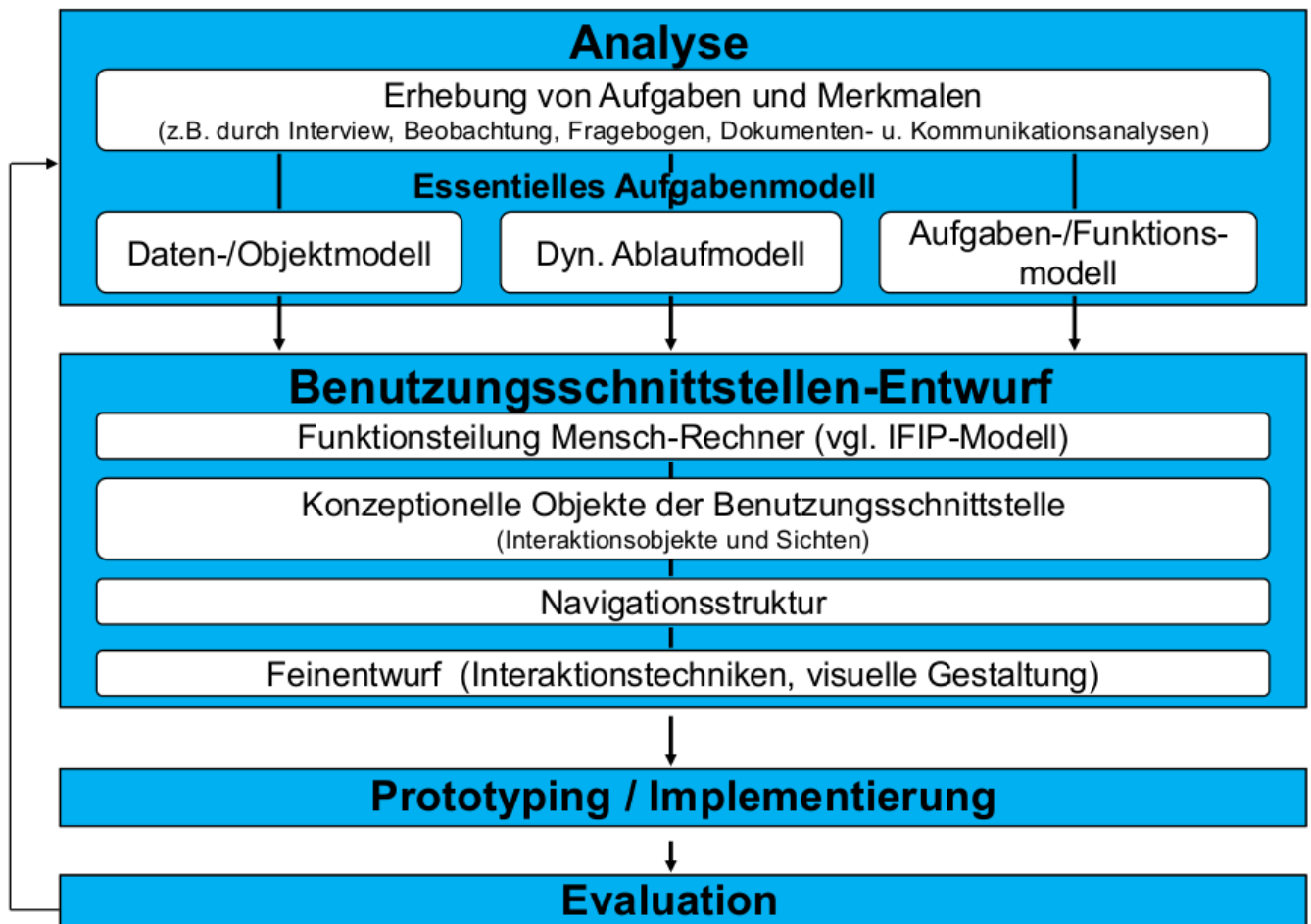
1. Konzeptuelle Ebene
2. Dialogebene
3. Ein-/Ausgabe-Ebene

Schichtenmodell

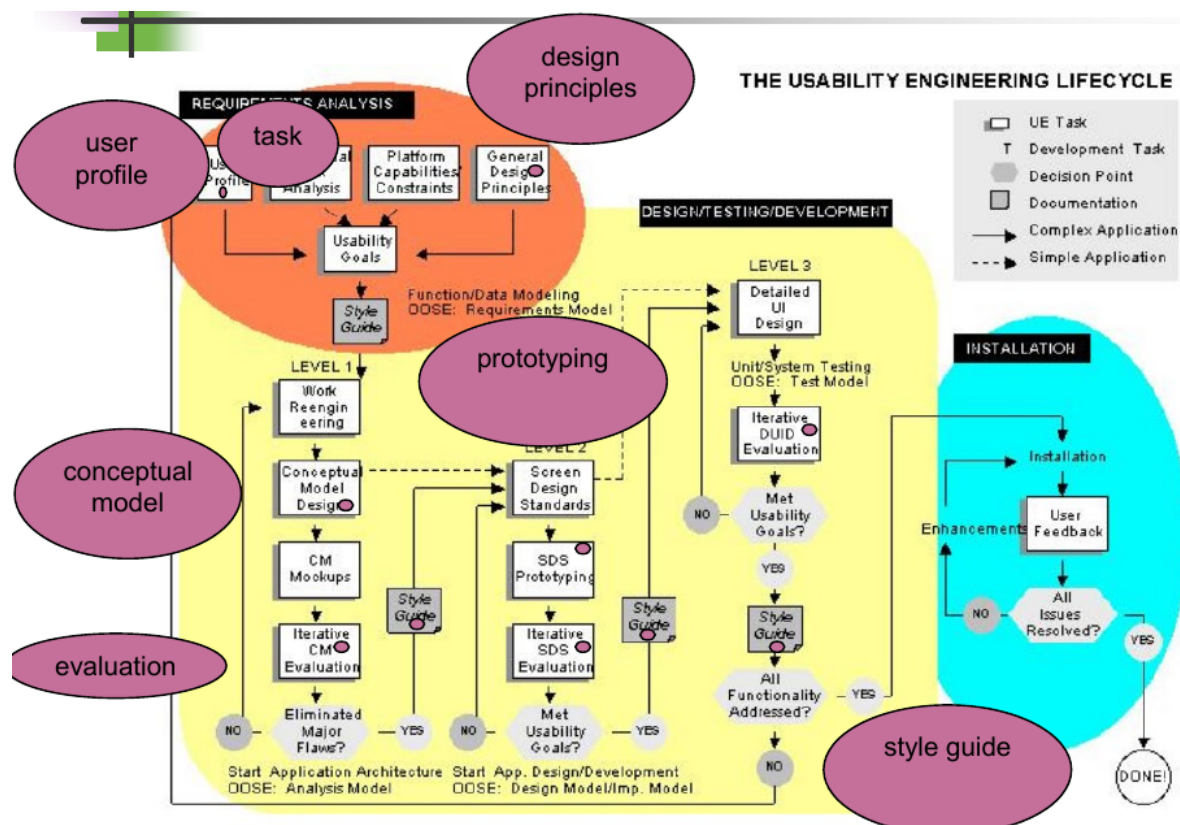


1.1.5 Erklären Sie User Centered Design.





1.1.6 Erklären Sie den Usability-Engineering-Lifecycle.



[Mayhew,01]

1.1.7 Welche Entwurfsprinzipien für MCI nach Preim kennen Sie?

1. Kenntnis potentieller Benutzer und ihrer Aufgaben
2. Unterstützung beim Aufbau mentaler Modelle
3. Terminologie der Benutzer verwenden
4. Reduktion der kognitiven Belastung
5. Strukturierung der Benutzungsoberfläche
6. Kombination visueller und textueller Elemente
7. Sichtbarkeit von Systemzuständen und möglichen Aktionen
8. Angemessene Rückkopplung
9. Konsistenz
10. Abbruch und Rückgängig machen von Aktionen
11. Erwartungskonformes Verhalten
12. Adaptierbarkeit der Schnittstelle

1.1.8 Wie sollten Dialoge laut ISO 9241-110 gestaltet werden? Beispiele?

1. Aufgabenangemessenheit
 - Abschließen einer Aufgabe
2. Selbstbeschreibungsfähigkeit
 - Jederzeit kann der Benutzer erkennen in welchem Dialog er sich befindet und was er wie erreicht
3. Steuerbarkeit
 - jederzeit den Fortgang initiieren
4. Erwartungskonformität
 - Dialog ist im Kontext und entsprechend den Konventionen ausführbar
5. Fehlerrobustheit
 - trotz Fehlern, evt. mit kleinen Änderungen, ist das Ziel erreichbar
6. Lernförderlichkeit
7. Individualisierbarkeit
 - Interaktion und Präsentation sollen den eigenen Anforderungen entsprechen
 - (Schriftgröße, Farben, Tastenkürzel, Anordnungen)

1.1.9 Was ist Usability? Wie kann sie bewertet werden?

- Gebrauchstauglichkeit
 - Effektivität (ich komme zum Ziel)
 - Effizienz (mit wie viel Aufwand komme ich zum Ziel)
 - Zufriedenheit (wie angenehm war der Weg zum Ziel)

1.1.10 Wie werden Benutzungsoberflächen evaluiert?

- Sammeln von Daten
- Beobachtung, Aufzeichnung
- Befragung
- Expertenmeinung
- Kommentare von Kunden
- 5-7 Nutzer um Entwurfsfehler zu finden

Technology Acceptance Model

Bewertung des Käuferverhaltens von Software.

1.1.11 Grenzen Sie die Begriffe Mentales Modell und Konzeptionelles Modell voneinander ab.

Konzeptionelles Modell

- vom Designer verfolgt und soll objektiv nachvollziehbar sein
- Metaphern sind bei Entwicklung hilfreich
- Beschreibung eines künftigen Systems von Begriffen in Ideen und Konzepten darüber
 - was es tun soll
 - wie es sich verhalten soll
 - wie es aussehen soll
 - verständlich für den Zielnutzer

Mentales Modell

Ein mentales Modell ist die Vorstellung die Personen von sich, anderen, ihrer Umgebung und den Dingen mit denen sie interagieren haben. Personen formen mentale Modell durch ihre Erfahrungen, Übung und Anleitung.

Mentales Modell ist die Vorstellung des Nutzers wie das System arbeitet.

1.1.12 Wie wird ein konzeptionelles Modell analysiert?

- welche Konzepte wird der Benutzer anwenden? (Metapher, Daten, Semantik?)
- gibt es vergleichbare andere Systeme?
- Art der Metapher (realitätsbasiert, naive Physik)
- relevante Interaktionstechniken?
- Fehlerursachen? (Rückgängig machen planen)

1.1.13 Wodurch entstehen fehlerhafte mentale Modelle?

Fehler entstehen beim Erheben der mentalen Modell und sind bedingt durch:

- fehlerhafte Wiedergabe/Verbalisierung durch den Benutzer
- fehlerhafte Interpretation durch den Designer (kulturelle Missverständnisse, Unkenntnis behinderungsspezifischer Anforderungen)

1.1.14 Welche Methode zur Aufgabenanalyse kennen Sie?

HTA ... Hierarchische Aufgabenanalyse (Hierarchical Task Analysis)

- betrachtet das beobachtbare Verhalten
- nicht sehr detailliert
- Grundlage für das Verständnis über den Benutzer (dessen Anforderungen, Ziele, Kenntnisse, usw.)

Bsp.:

1. um eine Kopie anzufertigen
 1. PIN eingeben
 2. Dokument mit Textseite nach unten einlegen
 3. Kopierdetails festlegen
 1. Anzahl der Kopien
 2. Papiergröße einstellen (Quelle/Ziel)
 4. Kopiertaste betätigen
 5. Kopie entnehmen

1.1.15 Wie wird bei einer Hierarchischen Taskanalyse vorgegangen?

1. Bestimmen der Nutzergruppen
2. Benennen der Hauptaufgaben
3. Vorbereitung und Durchführung einer Datensammlung über Aufgaben:
 - Ziele der Benutzer
 - damit verbundene Aktionen
 - Gründe für Aktionen
 - Verwendung von Informationsquellen
 - Interviews, ethnografische Experimente, Recherche
4. Datenanalyse, anfängliche Aufgaben, Stoppregeln (notwendige Tiefe der Aufgabenhierarchie)
5. Erstellen allgemeines Aufgabenmodell durch Verallgemeinerung der Aufgaben
6. Modellabstimmung mit Stakeholdern, Analytikern
7. Iterationen planen

1.1.16 Wo hat die HTA ihre Grenzen?

- nicht sehr exakt (z.B. warten)
- ungeeignet für die Interaktion von Benutzern untereinander
- wenig zum Kontext evt. auch zum Warum?
- manchmal zu sehr auf Methode fokussiert
- möglich, dass aktuelle Aufgabe zu sehr betont wird
- GUIs und OS für mehr als nur eine Aufgabe geeignet

1.1.17 Welche Informationen zum Kontext sind zu beachten?

- physikalisch (laut, staubig, hell, sicher, ...)
- sozial (Informationsweitergabe erwünscht, Privatsphäre, Kommunikation, ...)
- kulturell (Etikette, Lesart, Terminologie, Farben, ...)
- organisatorisch (hierarchisch, Managementstil, Benutzerunterstützung, ...)

1.1.18 Was bedeutet GOMS?

- **Goals**
 - Zielzustand den der Benutzer erreichen will
- **Operators**
 - Handlungen (z.B. Maus bewegen, Taste drücken)
- **Methods**
 - Reihenfolge der Operatoren für ein Ziel oder Teilziel
 - Beispiel: Tastenanschläge beim editieren
- **Selection Rules**
 - Anwendung bei mehreren Operatoren

GOMS zerlegt Benutzerinteraktion mit Computer in elementare Aktionen. Dies erlaubt Untersuchung von Benutzerschnittstellen.

Beispiel:

1. Maus nehmen .3s
2. Ziel suchen .2s
3. Maus zu Ziel bewegen .5s
4. Ziel anklicken .1s
5. Warten auf Programmstart 4s
6. Nachdenken/Überblick verschaffen 3s usw.

1.1.19 Nennen Sie die Phasen der Software-Entwicklung.

1. Analysephase
 - Aufgabenanalyse (HTA)
 - Kontextanalyse
 - Stakeholder, Szenarios, Personas
 - Claims
2. Designphase

- Activity Design
 - Information Design
 - Interaction Design
3. Entwicklungsphase
4. Einführungsphase

Prototyping findet in Phase 2 und 3 statt. Evaluation von Phase 2 bis 4.

1.2 Evaluation

1.2.1 Was verstehen Sie unter dem Begriff Ergonomie?

Ergonomie ist eine wissenschaftliche Disziplin, die sich mit dem Verständnis der Wechselwirkungen zwischen menschlichen und anderen Elementen eines Systems befasst und der Berufszweig, der Theorien, Prinzipien, Daten und Methoden auf die Gestaltung von Arbeitssystemen anwendet mit dem Ziel, das Wohlbefinden des Menschen und die Leistung des Gesamtsystems zu optimieren. (ISO 6385)

1.2.2 Was verstehen Sie unter dem Begriff Usability?

Usability oder *Gebrauchstauglichkeit* ist das Ausmaß, in dem ein technisches System durch bestimmte Benutzer in einem bestimmten Nutzungskontext verwendet werden kann, um bestimmte Ziele **effektiv**, **effizient** und **zufriedenstellend** zu erreichen. (ISO 9241, Teil 11)

1.2.3 Wann kann Evaluation in den Designprozess einwirken?

formativ ... während Designprozess

summativ ... nach Designprozess

1.2.4 Beschreiben Sie den Lebenszyklus beim Usability-Engineering.

- Analysephase
 - Arbeits-, Prozess- und Systemanalyse
 - Erhebung der Nutzeranforderungen
- Konzeptphase
 - Arbeitsgestaltung und Prozessdefinition
 - Entscheidung über Systemfunktionalitäten
 - Konzepterstellung (Mock-Ups)
- Entwicklungsphase

- Entwicklung von Prototypen
- Systemintegration
- Einführungsphase
 - Piloteinsatz
 - Allgemeine Einführung

1.2.5 Nennen Sie Methoden der Partizipation.

- Wer wird beteiligt? (Nutzer/Nutzervertreter mit Wissen zu Anwendungsdomäne)
- Wie wird beteiligt? (passiv, nach Ermessen der Entwickler; aktive Mitwirkung, gemeinsam mit Verantwortlichen; aktive Partizipation, Nutzer gestalten direkt)
- Wann wird beteiligt? (bestimmte Zeitpunkte oder dauernd)
- Woran wird beteiligt? (Definition, Gestaltung der Funktionen, Gestaltung Arbeitsumfeld)

1.2.6 Welche Nutzereigenschaften sind bei der Evaluierung zu berücksichtigen?

- Benutzergruppen inhomogen
 - unterschiedliche Erfahrung mit System
 - Erfahrung mit Computern
 - Erfahrung mit dem Aufgabenbereich
 - Alter, Geschlecht Farbenblindheit, Rechts-/Linkshänder, Intelligenz, räumliches Vorstellungsvermögen
- Benutzer verändern sich mit dem System
 - manchmal Lerneffekt während des Tests
 - starke Variationsbreite
 - schwer Modell vorherzusagen

1.2.7 Nennen Sie Testverfahren der Usability.

- Befragung
 - kontext-bezogene Befragung
 - Interviews
 - Umfragen
 - Fragebogen
 - kritische Vorkommnisse
 - aufgezeichnete Sitzungen
 - Eigenbericht (Logs)
 - Beispielbildschirme
- Beobachtung
 - Thinking Aloud
 - heuristische Evaluation
 - Cognitive Walkthroughs
 - Formale Usability Inspections

- Pluralistische Walkthroughs
- Feature Kontrolle
- Standards Kontrolle
- Checksite Richtlinien
- Performance-Messung (z.B. Eye-Tracking)
- Kreative Methoden
 - Fokusgruppen und Workshops
 - Theaterbasierte Evaluation
- Softwaretests
 - Modellbasierte Methoden
 - GOMS Analyse
 - KLM
 - MAD
 - automatisierte Evaluation

1.2.8 Welche Aspekte sind bei der Evaluation von Zeigehandlungen zu berücksichtigen?

- Anzahl der Dimensionen/Freiheitsgrade
- Position
- Art der Bewegung
- Art der Kraftanwendung

1.2.9 Erklären Sie Fitts Gesetz

- Die Zeit für das Erreichen eines Ziels ist abhängig von der Entfernung und der Größe des Ziels (Fitt, 1954).

$$MT = a + b \log_2(2A / W + c)$$

- MT ... Bewegungszeit (movement time)
- a und b ... empirisch bestimmte, geräteabhängige Konstanten
- c ... Konstante von 0, 0,5 oder 1
- A ... Entfernung (oder Amplitude) der Bewegung vom Startpunkt bis zum Zielmittelpunkt
- W ... Breite des Ziels, Genauigkeit

ID ... Index of Difficulty, $\log_2(2A / W + c)$ IP ... Index of Performance, $1/b$

Ds bedeutet, große Ziele in der Nähe werden schneller erreicht als kleine, weit entfernte Ziele.

1.2.10 Drücken Sie Fitts Gesetz kurz und knapp aus.

Je näher und/oder je größer ein Ziel ist, desto schneller kann es erreicht werden.

1.2.11 Nennen Sie Erweiterungen von Fitts Gesetz.

- **Status Quo** - nur Breite geht ein
- **Summenmodell** - $W = \text{Höhe} + \text{Breite}$
- **Flächenmodell** - $W = \text{Höhe} * \text{Breite}$
- **Smaller of - Modell** - $W = \min(\text{Höhe}, \text{Breite})$
- **W-Modell** W ist Länge von Schnittpunkt bis Mittelpunkt des Ziels

1.2.12 Wie kann man sich Fitts Gesetz zunutze machen?

- Ziel vergrößern (Fish Eye, großer Cursor, Standardauswahl vergrößern, ...)
- Distanz verringern (Objekte kommen entgegen, Kuchenmenüs, Platzierung, ...)
- beides

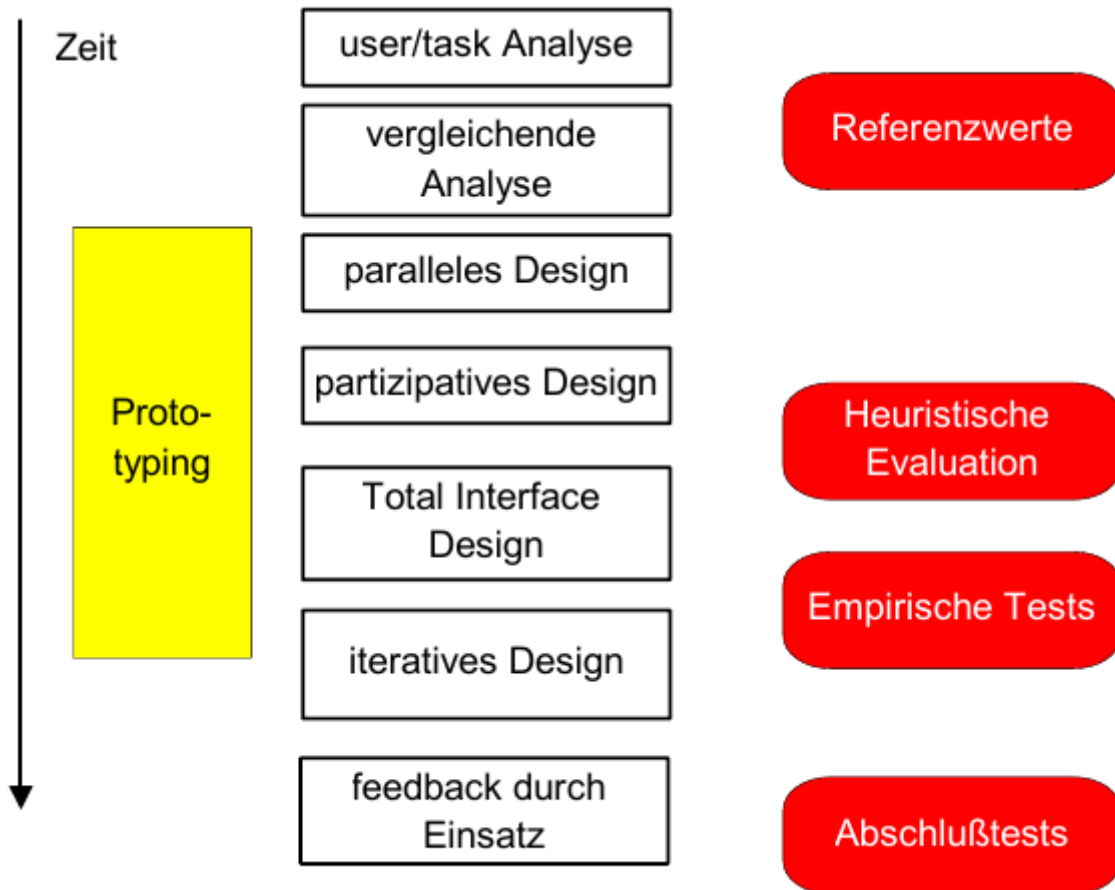
1.2.13 Womit befasst sich das Steering Law?

Wie Fitts Gesetz, aber nur für geradlinige (z.B. rein horizontale Bewegung) zwischen Begrenzungen.

1.2.14 Wo darf Fitts Gesetz nicht angewendet werden?

3D-Bewegungen auf 3D-Oberflächen, Sakkaden, ...

1.2.15 Wo im Lifecycle bei der Softwareentwicklung ist das Usability Engineering angesetzt?



1.2.16 Erklären Sie heuristische Evaluation

- Experten prüfen Programm auf Einhaltung vorgegebener Richtlinien
- jedes Modul mindestens zweimal getestet
- konstruktive Vorschläge mit Priorität (kosmetisch, klein, groß, katastrophal)

1.2.17 Nennen Sie die 10 Heuristiken nach Nielsen und Molich.

1. Sichtbarkeit des Systemstatus
2. Übereinstimmung zwischen System und realer Welt (z.B. Wortwahl)
3. Benutzerkontrolle und Freiheit
4. Konsistenz und Standards
5. Fehlervermeidung und Fehlertoleranz
6. Erkennen vor Erinnern
7. Flexibilität und effiziente Nutzung
8. Ästhetisches und minimales Design
9. Unterstützung beim Erkennen, Verstehen und Bearbeiten von Fehlern
10. Hilfe und Dokumentation

1.2.18 Was ist ein Cognitive Walkthrough?

- Evaluation für unerfahrene Nutzer

- Aufgaben im Umfang eines Tutorials für Nutzer
- Usability-Experten diskutieren vier Leitfragen:
 1. Werden Nutzer versuchen gewünschten Effekt zu erzielen?
 2. Werden Nutzer erkennen, dass korrekte Handlung ausgeführt werden kann?
 3. Werden Nutzer erkennen, dass korrekte Handlung zum gewünschten Effekt führt?
 4. Werden Nutzer Fortschritt bei korrekter Handlung erkennen?

1.2.19 Weshalb werden Prototypen benötigt?

- GUI meist erst spät in Entwicklung verfügbar
- Prototyp sehr früh zur Evaluation
- Prototyp schnell und billig (ok wenn: eingeschränkte Funktionen, nicht optimal effizient, instabil, unzuverlässig, vereinfacht, nur 1 Plattform, Pseudodaten, Wizard of Oz, Paper-Mockup)

1.2.20 Was ermittelt der NASA TLX Fragebogen?

Wie anstrengend (physisch, psychisch, zeitlich, emotional) die Durchführung einer Aufgabe war. Höher = schlechter

Mathematisch zweifelhaft, da Ordinalskala (anstrengen, weniger anstrengen, ... nicht äquidistant)

1.3 Fenstersysteme

1.3.1 Erklären Sie Interaktion

Begriffe

- Hintergrund: den Nutzer Unterstützen Ziele in einer bestimmten Aufgabendomäne zu erreichen
- Domäne: Bereich von Fachwissen und Kenntnissen in bestimmter reeller Aktivität
- Aufgabe: Operation zur Manipulation des Konzepts einer Domäne
- Ziel: gewünschtes Ergebnis einer Aufgabe
- Intention: bestimmte notwendige Aktion zum Erreichen eines Ziels
- Aufgabenanalyse: Untersuchung des Problemraums des Benutzers eines interaktiven Systems (Domäne, Ziele, Intentionen und Aufgaben)
- Systemsprache: Form der Kommunikation auf Systemseite
- Nutzersprache: Form der Kommunikation auf Nutzerseite
- System: Computer-Anwendung

Erklären Sie den Ausführungs-Evaluationskreislauf (Normans Modell)

- Nutzer formuliert Plan
- Computer führt Plan aus
- Nutzer evaluiert PC-Ausgabe und bestimmt nächste Aktion

Ausführung und Evaluation können aufgeteilt werden in:

1. Ziel setzen
2. Intention bestimmen
3. Abfolg von Aktionen festlegen
4. Aktion ausführen
5. Systemstatus wahrnehmen
6. Systemstatus interpretieren
7. Systemstatus mit Blick auf Ziel und Intentionen evaluieren

Was versteht man unter Ausführungslücke und Evaluationslücke?

Unstimmigkeiten zwischen Nutzer und System

Ausführungslücke (Gulf of Execution)

Unterschied zwischen von Nutzer formulierten Aktionen und vom System erlaubten Aktionen.

Evaluationslücke (Gulf of Evaluation)

Unterschied zwischen physischer Ausgabe des Systemstatus und der Erwartung des Nutzers.

1.3.2 Nennen Sie Ihnen bekannte Interaktionstechniken. Sortieren Sie nach Initiative beim Computer bis Initiative beim Menschen.

Initiative eher beim Computer

- Kommandozeile
- Menü
- natürliche Sprache
- Frage/Antwort-Dialog, Abfragesprachen
- Formulare, Tabellen
- WIMP
- Zeigen, Klicken
- 3D UI

Initiative eher beim Menschen

1.3.3 Unterscheiden Sie zwischen modalen und nichtmodalen Dialogen

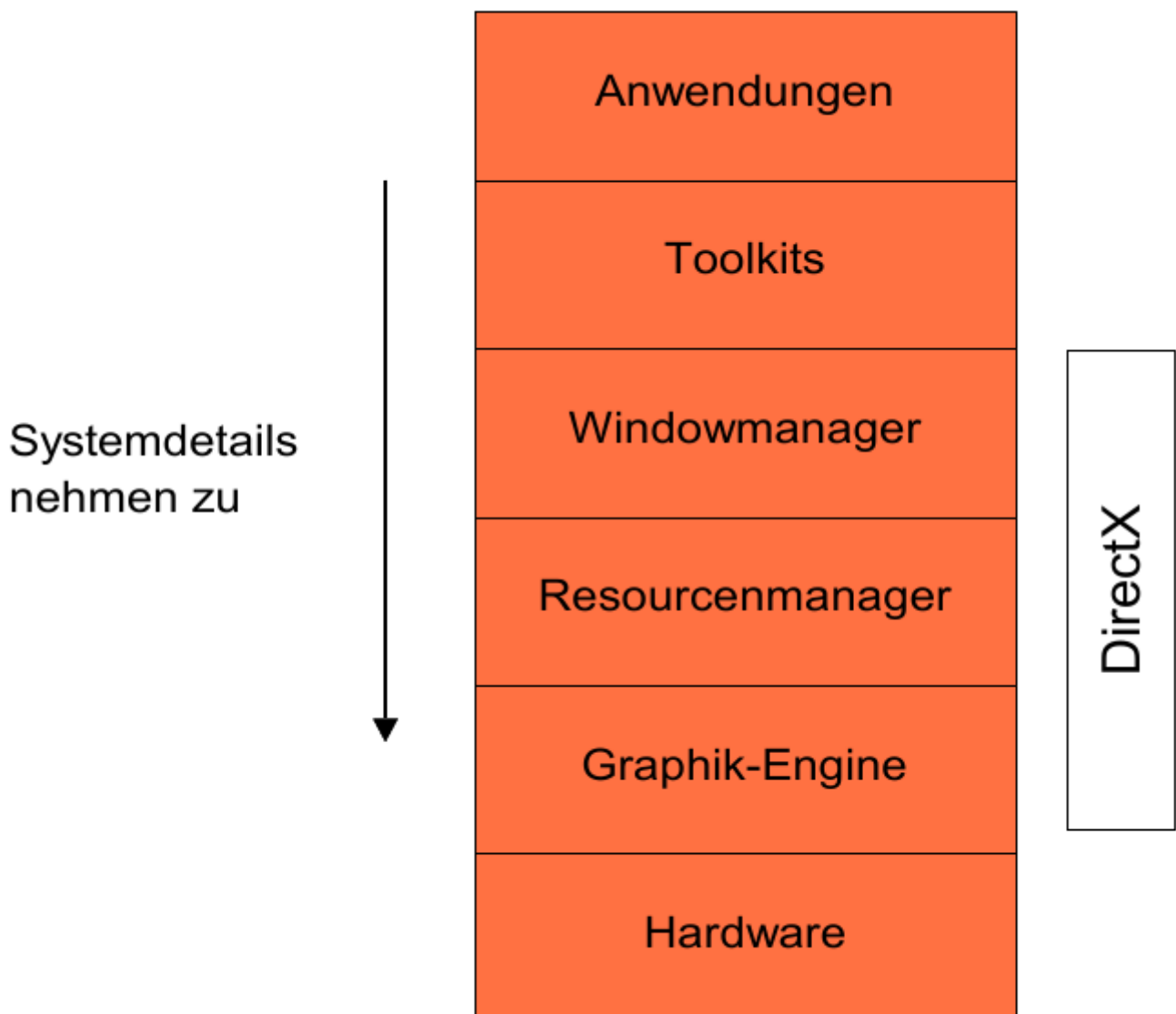
modal erzwingt Abschluss des Dialogschritts, blockiert restliches Programm

nichtmodal erlaubt parallele Bearbeitung (z.B. Suche, Rechtschreibprüfung)

1.3.4 Was versteht man unter Direkter Manipulation?

- alle relevanten Objekte sichtbar
- Input/Output gehen ineinander über
- auch kleinste inkrementelle Aktion gibt Rückmeldung (Bewegung, Töne usw.)
- immer Rückgängig möglich
- syntaktische Korrektheit aller Aktionen
- Vermeidung komplexer Kommandos
- Beispiele: Drag&Drop, Fenstersystem

1.3.5 Geben Sie die Komponenten eines Fenstersystems von hardware-fern zu hardware-nah an.



1.3.6 Nennen Sie wichtige Objekte der Ressourcenverwaltung

- Fenster
- Ereignisse
- Queue
- Prozess/Instanz
- Multiplexing
- Schriftarten
- Grafikkontexte
- Farbtabelle
- Fernzugriff

1.3.7 Welche Aktionen auf Ressourcen werden ausgeführt?

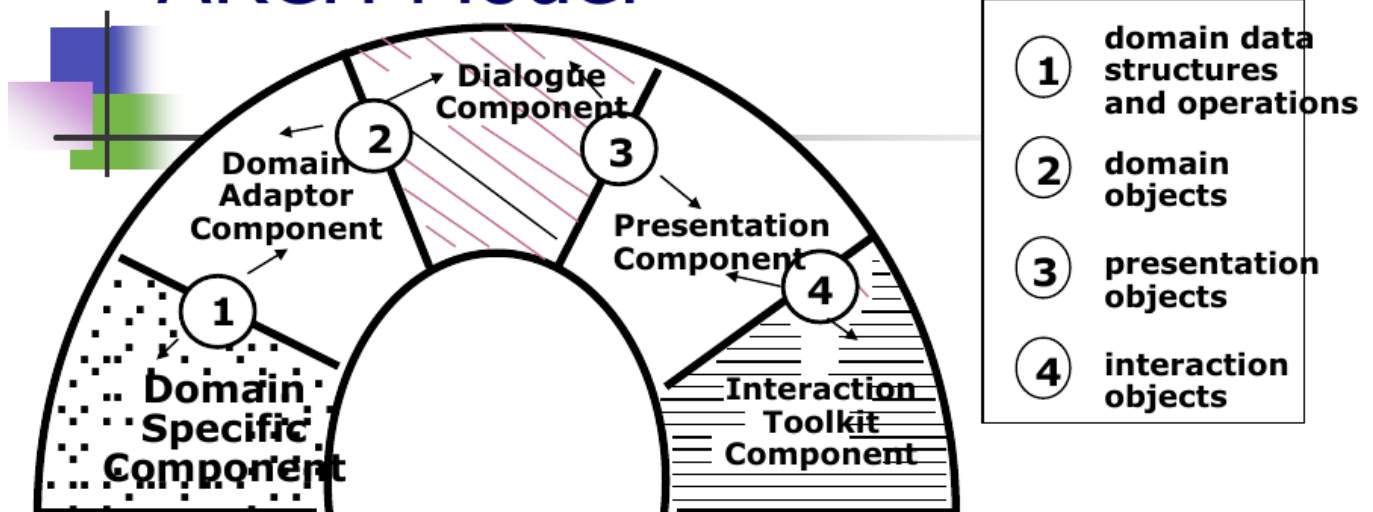
- Allokation / Deallokation
- Initialisierung
- Synchronisation

1.3.8 Worum kümmert sich ein Fenstermanager?

- Fokus und Verdeckung
- logischer Bildschirm / physische Bildschirme
- Platzierung
- Fenstermenüs
- Rahmen, Kachelung/Überlappung
- Zeichnen
- Drag'n'Drop
- Maus
- Listeners / Events
- Look and Feel

1.3.9 Erklären Sie das ARCH Model

ARCH Model



The ARCH Model for User Interface Environments

(Seeheim revisited, User Interface Developer's Workshop 1991)

1.3.10 Sortieren Sie die Erstellung von GUIs nach Abstraktionsgrad von niedrig zu hoch.

- Programmierung (von Hand in Programmiersprache mit Bibliotheken/Toolkits)
- textuelle Spezifikation (in spezieller Beschreibungssprache, zB XML)
- strukturiertes Editieren (grafischer Editor, syntax-sensitiver Editor)
- automatische Generierung

1.3.11 Was sind Application Frameworks?

- enthalten Klassen für Dialogobjekte (Warnhinweis, Dateiauswahl, ...)
- generische Klassen für Entwicklung der Hauptanwendung
- Beispiele: Java AWT, Java FX, Swing, Gtk+, Qt
- Persistenz (z.B. als XML-Datei) ist möglich
- Bausteine für Oberflächenobjekt
- Style-Guide-Unterstützung
- Datenstrukturen
- Erweiterbarkeit
- komplex

1.3.12 Was sind User Interface Builder?

- Erlauben Beschreibung einer Oberfläche mittels einer Oberflächenbeschreibungssprache
- auch für grafische Oberflächeneditoren
- Trennung von UI und Restprogramm

1.3.13 Was sind User Interface Management Systeme?

- Erweiterung von UI-Beschreibungssprachen
- erlauben Beschreibung von Dialogabläufen
- Steuern von Ein- und Ausgabegeräten
- Bereitstellung Hilfsinformationen
- Plausibilitätstests
- Auffrischung des View
- Benachrichtigung bei Viewänderung und Adaption an Klassen
- Nutzung darunterliegender Dienste
- Simulation

1.3.14 Welche Vorteile haben UIMS?

- Vereinfachung der Programmierung
- Anwendung und GUI unabhängig voneinander entwickelt
- hohe Flexibilität und Wartungsfreundlichkeit der GUI
- einfache Lokalisierung
- einfache Wiederverwendung
- vereinfachte Evaluation durch Simulation

1.3.15 Wie funktionieren automatisch generierende Werkzeuge?

- Generierung aus abstrakter Spezifikation
- verwendet definierte Transformationsregeln
- Navigationsstruktur > Dialogspezifikation
- plattformunabhängiges UI-Metamodell
- nachträgliche Modifizierung

1.4 Formale Modelle und Zeit

1.4.1 Nennen Sie die Ziele formaler Modelle

- Analyse UI auf bestimmte Eigenschaften
- Anwenden Modelle menschlicher Performanz (GOMS)
- automatische Kritik zum Design
- Beweis sicherheitskritischer Aspekte

- Ermittlung kürzester Wege (Floyd-Warshall) durch Matrix der Tastenfolgen

1.4.2 Welche Phasen der Interaktion bestimmen das Erleben der Interaktion?

- Eingabezeit
- Antwortzeit
- Ausgabezeit
- Denkzeit

1.4.3 Was beschreiben temporale Modelle (z.B. Petrinetz, Semaphore Intervalldiagramme)?

- zeitliche Strategie des Benutzers während der Interaktion
- zeitliche Beschränkungen während der Interaktion
- Nebenläufigkeit von Interaktionsschritten

einer Benutzungsoberfläche.

1.4.4 Was ist ein Intervalldiagramm (Weber)?

Beschreibt durch temporale Intervall die Lebensdauer eines Objekts oder die Ausführbarkeit von Operationen.



1.4.5 Grenzen Sie Intervalle und Ereignisse ab.

- Ereignisse sind Kontaktpunkte
- Ereignisse sind selbst Intervalle
- direkte Kontaktpunkte haben Bezug zu beiden Medien
- indirekte Kontaktpunkte stellen Bezug zu vorhergehender Referenz in anderem Medium her
- Gestaltungsraum für Kontaktpunkte für jedes Medium unterschiedlich

1.4.6 Wie kann man die Wartezeit tolerierbar gestalten.

- Fortschrittsanzeige (bei >0% anfangen)
- mehr Zeit als berechnet angeben (positive Überraschung)

- letzten Abschnitt genaue Zeitangaben
- Überbrückung durch extra Informationen
- abnehmende Intervalle
- Ablenkung
- Einmal-Effekt (bei einmaligen Aktionen)

1.5 Adaptierung und Adaptivität

1.5.1 Was versteht man unter Adaptierung?

- Veränderung der Einstellungen des Programms durch Anpassung

1.5.2 Was versteht man unter Adaptivität?

- selbständige Anpassung des Systems and Eingaben oder von Ausgaben

1.5.3 Nennen Sie Bestandteile eines adaptiven Systems.

- Sammeln von Informationen über Nutzer
- Auswertung der Information
- Anpassen des Systems anhand der Information

1.5.4 Erläutern Sie die Methodik adaptiver Systeme.

1. Afferenz - Beobachtung und Sammlung von Informationen
2. Inferez - Auswerten der gesammelten Informationen
3. Efferenz - Anpassung des Systems

1.5.5 Welche Daten sammelt so ein System über den Nutzer?

- Wissensstand des Nutzers
- Pläne und Ziele des Nutzers
- Benutzervorlieben
- Benutzerfähigkeiten
- Benutzermissverständnisse

1.5.6 Welche Formen der Adaptivität gibt es beim Information Retrieval?

- adaptive Führung: z.B. Auszeichnen relevanter Verweise
- adaptive Annotation: z.B. Generierung von Hinweisen
- adaptive Empfehlung: z.B. Vorauswahl und Anzeige interessanter Inhalte

1.5.7 Was versteht man unter dem Bewertungsproblem?

- Ermitteln der Bewertung von Objekten die ein Nutzer nicht kennt, z.B. zum Vorschlagen.
- inhaltsbasiert vs. kollaborativ vs. hybrid

1.6 Interaktionsdesign und Kontext

1.6.1 Was sind Dialogbausteine?

Sind vordefinierte Interaktionselemente graphischer UI-Elemente, für die Aussehen (Attributwerte) und Verhalten (Reaktion auf Ereignisse) bereits festgelegt sind.

- entlasten Entwickler
- wiederverwendbar, standardisiert
- erlauben Erstellen von Beispielen
- Qualitätssicherung vereinfacht
- erprobt
- verringerte Iterationszyklen beim Prototyping
- einheitliches Aussehen

1.6.2 Was ist ein Pattern?

Ist eine bewährte Lösung zu häufigen Problemen in einem gegebenen Zusammenhang, hier für MCI. *Pattern Language* definiert Sammlung von Mustern und Regeln für die Kombination zu Architektur-Stilen.

Eigenschaften:

- Wiederholbarkeit
- Bewährtheit
- Abstraktion
- Rahmenbedingungen
- Identifikation

1.6.3 Aus welchen Ebenen besteht das UI-Pattern-Konzept?

- Portal Floorplan/Navigation (Aufsuchen vordefinierter Grundrisse, z.B. Rechnung bearbeiten, Zähler, ...)
- Aktivitätsgrundrisse (Auffindungsmuster, Datenstrukturen)
- UI-Patterns (show, get, select)
- Pattern-Elements (Dropdown-Box, Suchfeld, ...)
- UI Controls (Buttons, Eingabefeld, Tabelle, Baum, ...)

1.6.4 Nennen Sie einige UI Patterns.

- kontinuierlicher Filter
- doppelte Liste
- Fish-Eye
- Überblick und Detail
- Globale Navigation

1.6.5 Was versteht man unter dem Kontext im UI Design?

- Umgebung (Temperatur, Licht, Krach, ...)
- soziale Aspekte (Privatsphäre, Zusammenarbeit, ...)
- Nutzer (Vorlieben, Ziele und Aufgaben, physischer & emotionaler Zustand, ...)
- Technik (Geräte, Browser, Internetverbindung, ...)

1.6.6 Was ist kontextuelles Design?

- kundenzentrierter Ansatz (Feldstudie)
- ethnografischer Ansatz, Entwicklung aus Nutzersicht mit Nutzerbeteiligung
- Unterstützung der Arbeitsgruppen und Organisationen

Modelle:

- Rollen und Austauschmodelle
- Ablaufmodelle
- Artefakt-Modelle (z.B. Artefakt persönlicher Kalender)
- Einfluss-Modelle
- physische Modelle (z.B. Struktur des Raums)

1.6.7 Was ist context-aware-computing?

Software reagiert auf sich ändernden Kontext mit Anpassung. (Bildschirmhelligkeit, Lautstärke usw.)

1.6.8 Wie kommt es zum Awareness-Mismatch?

System passt sich anders an als Nutzer es erwarten würde.

Beispiel: Display schaltet sich aus, wenn Nutzer Handy ans Ohr hält, Nutzer versucht aufzulegen, aber Display inaktiv.

1.7 8

1.7.1 Was ist ein ConcurTaskTree?

CTT ... Concur Task Tree

Basis für den Austausch von Hochstufigen UI-Aufgabenmodellen.

- Fokussierung auf Aktivitäten, kein verheddern in Details
- hierarchische Struktur, Aufspalten von Problemen in kleinere Probleme bei Erhalt der Beziehungen, Wiederverwendbarkeit
- grafische, leicht zu interpretierende Syntax in Baum-Form
- großes Arsenal an zeitlichen/temporalen Operatoren zur Definition von Abhängigkeiten und Beziehungen
- Aufgabenzuweisung
- Objekt- und Aufgabeneigenschaften

1.7.2 Wofür steht GTA?

GTA ... Groupware Task Analysis

Vereinigung von CSCW- (Computer Supported Collaborative Work) und MCI-Design.

Methode zur Aufgabenanalyse die sich mit dem Kontext der Nutzung eines Systems im weitesten Sinne beschäftigt. Dabei wird die Welt der Aufgabe aus drei Blickwinkeln (Agent (Nutzer, Nutzergruppen, kooperierende Nutzergruppen), Arbeit (Aktivitäten, Ziele, kooperative Aktivitäten), Situation) betrachtet.

Agents

- menschlich oder nicht-menschlich
- Actor: meist Individuen, psychologische Charakteristika (kognitiver Typ, räumliches Vorstellungsvermögen), Fachwissen
- Rolle: Klasse von Actors der verschiedenen Aufgabentypen zugewiesen sind, ein Actor kann mehrere Rollen haben
- Organisation: Verhältnis zwischen Actors und Rollen im Bezug auf Aufgabenverteilung, Zuweisen von Aufgaben von einer Rolle zur anderen

Arbeit

- Ziele, Aufgaben, viele Aufgaben können gleiches Ziel haben, jede Aufgabe hat genau ein Ziel

Situation

- Erkennen und Beschreiben der Umwelt (physisch, sozial usw.)
- Objekte, Objektstruktur

1.7.3 Was versteht man im Zusammenhang mit MCI unter

Ethnographie?

Unter Ethnografie versteht man die Ermittlung der Sicht des zukünftigen Nutzers auf einen Sachverhalt der durch eine zu entwickelnde Software behandelt werden soll.

Dadurch sollen Anwenderbedürfnisse ermittelt werden. Software-Ergonomen setzen ethnografische Methoden ein mit dem Ziel der Veränderung und Verbesserung von Computerunterstützung.

1.7.4 Was versteht man im Zusammenhang mit MCI unter Ergonomie?

Ziel der Ergonomie ist es, die Arbeitsbedingungen, den Arbeitsablauf, die Anordnung der zu greifenden oder zu erreichenden Werkzeuge und Gegenstände räumlich und zeitlich optimiert anzuordnen sowie die Arbeitsgeräte auf eine Aufgabe so zu optimieren, dass das Arbeitsergebnis optimal wird und die arbeitenden Menschen bestmöglich unterstützt.

Besonderer Augenmerk auf Nutzerfreundlichkeit.

1.7.5 Was ist GroupWare?

GruppenSoftware, Kollaborative Software

1.7.6 Was sind XForms?

Plattformunabhängiger Standard für elektronische Formulare zur Datenerfassung.

- Model View Controller Architektur
- kaum von Browsern unterstützt
- XML-Grammatik

1.7.7 Was ist eine Ontologie?

Ontologien in der Informatik sind meist sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begrifflichkeiten und der zwischen ihnen bestehenden Beziehungen in einem bestimmten Gegenstandsbereich. Sie werden dazu genutzt, *Wissen* in digitalisierter und formaler Form zwischen Anwendungsprogrammen und Diensten auszutauschen. Wissen umfasst dabei sowohl Allgemeinwissen als auch Wissen über sehr spezielle Themengebiete und Vorgänge.

1.8 Mensch-Computer-Interaktion nach Alan Dix

1.8.1 Erklären Sie den Aufbau des menschlichen Gedächtnis.

Es besteht aus 3 Typen

- sensorischer Puffer
- Kurzzeitgedächtnis (Arbeitsgedächtnis)
- Langzeitgedächtnis

1.8.2 Beschreiben Sie das sensorische Gedächtnis.

Funktioniert als Puffer für Reize aller Sinne:

- Sehen: ikonisches Gedächtnis
- Hören: widerhall-Gedächtnis (echoic Memory)
- Fühlen: haptisches Gedächtnis

Informationen in diesen Puffern werden ständig durch neue Informationen überschrieben.
Aufmerksamkeit überführt Information vom sensorischen ins Arbeits-/Kurzzeitgedächtnis.

1.8.3 Beschreiben Sie das Kurzzeitgedächtnis.

- speichert Informationen die nur kurzzeitig gebraucht werden
- schnell abrufbar
- verblasst schnell
- geringe Kapazität: 5-9 Informationsblöcke (Einzelninformationen oder Gruppen von Informationen, zb 7 Zahlengruppen oder 7 Einzelziffern)
- Unterbrechung durch andere Informationen zerstört ältere Informationen

1.8.4 Beschreiben Sie das Langzeitgedächtnis.

- unbeschränktes Fassungsvermögen
- langsamer Zugriff
- langsames oder kein Verblässen von Information
- Information wird vom Kurzzeitgedächtnis durch Wiederholung hierher übertragen
- 2 Typen:
 - Episodisches Gedächtnis (Ereignisse, Erfahrungen in chronologischer Form)
 - Semantisches Gedächtnis (als Netzwerk strukturiertes Wissen von Fakten, Fertigkeiten usw.)
 - hierarchische Speicherung von generell zu speziellem Wissen (Lebewesen > Säugetier > Wolf)

Man kann 3 Hauptfunktionen des Langzeitgedächtnis unterscheiden:

- Speichern: Übertragen von Informationen aus Arbeitsgedächtnis durch Wiederholen, Anwenden usw., muss nützlich sein und sich integrieren lassen, je häufiger benutzt, desto besser abrufbar
- Vergessen: verschiedene Theorien (neues überschreibt altes Wissen / altes blockiert neues Wissen)
- Abrufen:
 - Erinnern: Gedächtnis wird mit Anhaltspunkten durchsucht und Informationen abgerufen
 - Erkennen: Information kann schneller abgerufen werden durch bekannte Darstellung der Information

1.8.5 Interaktion

1.8.6 Was ist Interaction Design?

Behandelt wie das erstellte Artefakt die Arbeitsweise von Personen beeinflusst.

1.8.7 Was ist Design?

- Design: Ziele unter Einschränkungen erreichen
- Ziele: die Absicht des zu erstellenden Designs
- Einschränkung: Einschränkung des Designprozess durch äußere Einflüsse
- Kompromiss: Wahl welche Ziele vernachlässigt werden können um andere zu erreichen

1.8.8 Was ist die goldene Regel des Design?

Verstehe dein Material: *Computer (Grenzen, Kapazitäten, Werkzeuge, Plattformen)* Mensch (Psyche, soziale Aspekte, Fehler)

1.8.9 Was versteht man unter *irren ist menschlich*?

Es liegt in der Natur des Menschen Fehler zu machen. Ein System muss so gestaltet werden, dass die Wahrscheinlichkeit von Fehlern und deren Konsequenzen minimiert werden.

1.8.10 Was sollte das Zentrum der Betrachtung beim Entwurf eines Systems sein?

Der Nutzer.

1.8.11 Umreißen Sie den Entwurfsprozess.

- Voraussetzungen: Beobachtung und Befragung zum Abbilden zu implementierender Funktionen
- Analyse: Schlüsselfunktionen durch Analyse der ermittelten Voraussetzungen
- Design: Umsetzen von Richtlinien
- Iteration und Prototyping: Evaluation früher Versionen mit echten Nutzern

- Implementierung und Veröffentlichung: Code schreiben, dokumentieren und Hardware

1.8.12 Wie kann man die Entwicklung auf den Nutzer konzentrieren?

- so viel Informationen wie möglich über den zukünftigen Nutzer sammeln
- Stakeholder: Personen die direkt oder indirekt vom neuen System betroffen sind
- Partizipatorisches Design: den potentiellen Nutzer komplett in den Designprozess einbinden
- Persona: detaillierte Beschreibung von imaginären Nutzern

1.8.13 Was ist in MCI ein Szenario?

Eine Kurzgeschichte für das Design welche detaillierte Angaben zur Interaktion mit dem System enthält.

1.8.14 Was verstehen Sie unter Usability Engineering?

Der Schwerpunkt beim Usability Engineering liegt darauf zu wissen, welche Kriterien die Usability eines Produkts bestimmen.

Für jedes Attribut gibt es sechs Punkte die die Nutzbarkeit des Attributs beschreiben:

- Measuring Konzept: Beschreibung des abstrakten Attributs in Worten des Produkts
- Measuring Methods: beschreibt wie die Attribute gemessen werden
- Now-level: aktueller Stand der Technik
- Worst case: niedrigster annehmbarer Wert für eine Aufgabe
- Planned Level: geplantes Design Level
- Best Case: bester möglicher Wert

Das Problem besteht darin in frühen Phasen geeignete Messwerte (Anzahl Knopfdrücke, Anzahl Schritte usw.) zu finden.

1.8.15 Welche 3 Prototyping-Ansätze kennen Sie?

- **Wegwerf-Prototyp:** durch Prototyp erhaltene Informationen werden verwendet, aber Prototyp selbst wird verworfen
- **Inkrementell:** das fertige Produkt ist ein Paket aus Komponenten die separat mit Prototypen entwickelt wurden
- **Evolutionär:** der Prototyp wird mit jeder Iteration des Designs verbessert und weiter entwickelt

1.8.16 Welche Probleme treten beim Prototyping auf?

Prototypen werden mit realen Nutzern getestet.

- Zeit: Prototyping kostet Zeit, welche beim richtigen Design fehlt -> deswegen Nutzung von Rapid-Prototyping-Techniken
- Planung
- Nicht funktionierende Teile: einige wichtige Features lassen sich mit Prototyp nicht testen
- Verträge: Prototyping kann nicht Basis eines rechtlichen Vertrages sein und braucht Dokumentation

1.8.17 Nennen Sie Prototyping-Techniken.

- Storyboards: grafische Vorschau des geplanten Systems, ohne Funktion
- beschränkte Simulation von Funktionen (vereinfachte Funktionen, scheinbare Funktionalität, Wizard of Oz)
- High-Level Programming Support: getrennte Entwicklung von Oberfläche und darunter liegender Funktionalität

1.8.18 Evaluation

Welche Bedeutung hat Evaluation bei der Entwicklung von Software?

- sollte den ganzen Entwicklungszyklus durchdringen
- Einbringung von Erkenntnissen in das Design
- Unterscheidung zwischen Evaluation von Entwicklern und Evaluation durch Nutzer

Was sind die Ziele der Evaluation?

- Erreichbarkeit der Funktionen des Systems abschätzen
- Erfahrung des Nutzers mit der Interaktion abschätzen
- bestimmte Probleme des Nutzers mit dem System aufdecken

Welche Methoden stehen Experten zur Evaluation zur Verfügung?

- Kognitiver Durchgang (cognitive Walkthrough):
 - detaillierte Betrachtung einer notwendigen Abfolge von Schritten für eine bestimmte Aufgabe
 - Ermitteln ungünstiger Punkte für neue Nutzer
 - Voraussetzung: Spezifikation oder Prototyp, Beschreibung der Aufgabe, komplette Liste notwendiger Schritte für bestimmte Aufgabe, Information über Nutzer
 - für jeden Schritt: Entspricht Effekt einer Aktion der Erwartung des Nutzers? Ist verfügbare Aktion sichtbar? Erkennen der Aktion als die richtige? Feedback?
- Heuristische Evaluation:
 - Beachtung von Richtlinien bei Designentscheidungen
 - Ermittlung ob Richtlinien berücksichtigt wurden
 - Bewertung: Wie oft tritt Problem auf? Wie leicht ist es zu lösen? Kurzzeitiges oder dauerhaftes Problem? Wie ernst wird Problem aufgefasst?
- Modellbasiert
- Studien ähnlicher Experimente wiederverwenden

Welche Methoden zur Nutzerzentrierten Evaluation stehen zur Verfügung?

Es müssen Laborstudien und Feldstudien unterschieden werden.

- **Beobachtung**
 - Laut Denken und Kooperative Evaluation
 - Aufzeichnung und nachträgliche Analyse
 - nachträgliche Befragung des Nutzers zu seinen Aktionen
- **Befragung**
 - Interviews
 - Fragebögen (mit Pilotstudie)
- **Verhaltensbeobachtung**
 - Blickverfolgung
 - Emotion durch Gesichtsausdruck
 - Herzschlag, Anspannung, Schwitzen usw.

1.8.19 Zählen Sie universelle Design-Prinzipien auf.

1. universell einsetzbar, unabhängig von den Fähigkeiten des Nutzers
2. Flexibilität (Anpassung an User, manuell und automatisch)
3. Einfach und intuitiv zu nutzen, Feedback
4. leicht aufzufassende Information
5. Fehlertoleranz und -Vermeidung
6. geringe Anstrengung (physische und psychische Erschöpfung)
7. Größe und Platzierung so, dass jeder Nutzer uneingeschränkt Zugriff hat

1.8.20 Was verstehen Sie unter multi-modaler Interaktion?

- Interaktion zwischen Mensch und Computer unter Nutzung mehrerer sensorischer Kanäle (sehen, hören, fühlen, ...)
- Ton: Spracherkennung, Sprechererkennung, Sprachsynthese, Signaltöne
- Touch: z.B. Blindenschrift, haptisches Feedback, vibrieren
- Handschrifterkennung
- Gestenerkennung

1.8.21 Welche menschlichen Unterschiede müssen Sie berücksichtigen und wie?

- Sehbehinderung: Hören, Fühlen
- Hörbehinderung: Fühlen (Vibrationsalarm)
- Körperliche Behinderung: Spracheingabe, Blickverfolgung
- Sprachbehinderung: Sprachsynthese
- Dyslexie: einfache Sprache, Sprachein- und -Ausgabe
- Autismus: kaum Probleme, da Computer *nicht soziales* Wesen
- Ältere: klare Formulierung, große Schriften, Fehlertoleranz
- Kinder: einfache Sprache, einfache Eingabe
- Kultur: Farbbedeutung, Symbole, ...

1.8.22 Worauf ist ein Nutzer bei Software-Hilfe angewiesen?

- Erreichbarkeit der Hilfe zu jeder Zeit
- Korrektheit und Vollständigkeit
- Konsistenz
- Robustheit
- Flexibilität
- soll Nutzer nicht an seiner Aufgabe hindern

2 Prüfungsaufgaben

Die hier aufgeführten Aufgaben wurden direkt in der mündlichen Prüfung gestellt. Es wurde erwartet, dass die Antwort detailliert ausgeführt und der Sachverhalt komplett verstanden wurde.

Die mündliche Prüfung erfolgt mit Professor und Beisitzer, sollte man nicht weiter kommen wird vorerst zur nächsten Frage übergegangen. Die Prüfdauer beträgt 25min.

2.1 Fitts Law

2.1.1 Erklären Sie Fitts Law

2.1.2 Schreiben Sie die Formel zu Fitts Law auf und erklären Sie Variablen und Konstanten

2.1.3 Erklären Sie die Bestimmung der Konstanten a und b mittels Messung von MT und A , sowie W

2.2 GPS

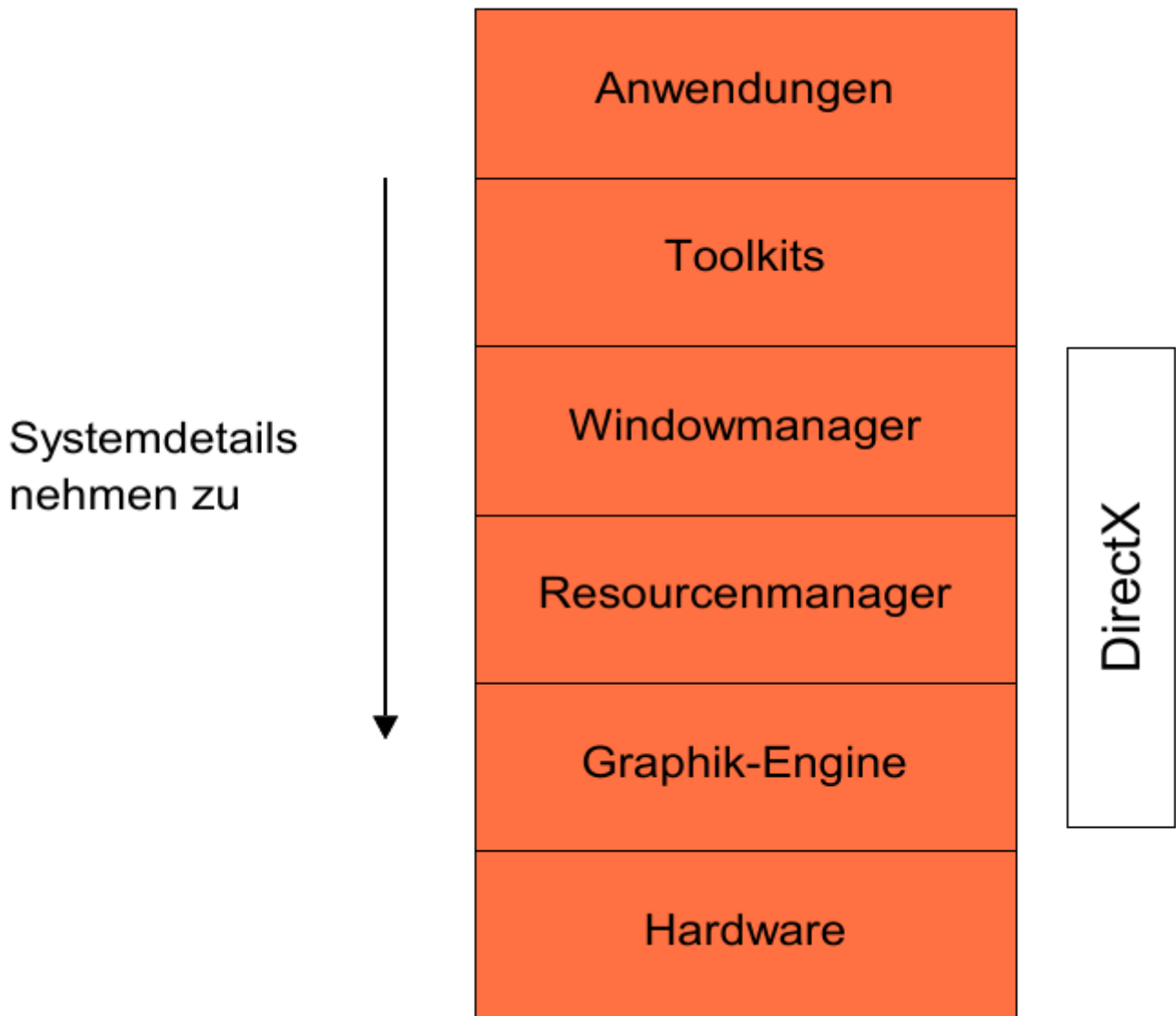
2.2.1 Erklären Sie die Funktionsweise von GPS und die Bedeutung aller 4 Satelliten

2.2.2 Nennen Sie andere Satelliten-Ortungstechniken

2.3 WiFi-Verortung

2.3.1 Erklären Sie die Funktionsweise von Ortsbestimmung mittels WiFi-Fingerprinting in Gebäuden

2.4 Schichtenmodell



2.4.1 Erklären Sie das Schichtenmodell mit Anwendungen, Toolkits, Fenstermanager, Ressourcenverwaltung, Grafik-Engine und Hardware

2.4.2 Erklären Sie die Verbindungen zwischen diesen Ebenen und auch die zwischen weiter voneinander entfernten Ebenen

2.5 Wie sollten Dialoge laut ISO 9241-110 gestaltet werden?

1. Aufgabenangemessenheit
 - Abschließen einer Aufgabe
2. Selbstbeschreibungsfähigkeit
 - Jederzeit kann der Benutzer erkennen in welchem Dialog er sich befindet und was er wie erreicht
3. Steuerbarkeit
 - jederzeit den Fortgang initiieren
4. Erwartungskonformität
 - Dialog ist im Kontext und entsprechend den Konventionen ausführbar
5. Fehlerrobustheit
 - trotz Fehlern, evt. mit kleinen Änderungen, ist das Ziel erreichbar
6. Lernförderlichkeit
7. Individualisierbarkeit
 - Interaktion und Präsentation sollen den eigenen Anforderungen entsprechen
 - (Schriftgröße, Farben, Tastenkürzel, Anordnungen)

2.6 Was versteht man unter Direkter Manipulation?

- alle relevanten Objekte sichtbar
- Input/Output gehen ineinander über
- auch kleinste inkrementelle Aktion gibt Rückmeldung (Bewegung, Töne usw.)
- immer Rückgängig möglich
- syntaktische Korrektheit aller Aktionen
- Vermeidung komplexer Kommandos
- Beispiele: Drag&Drop, Fenstersystem