

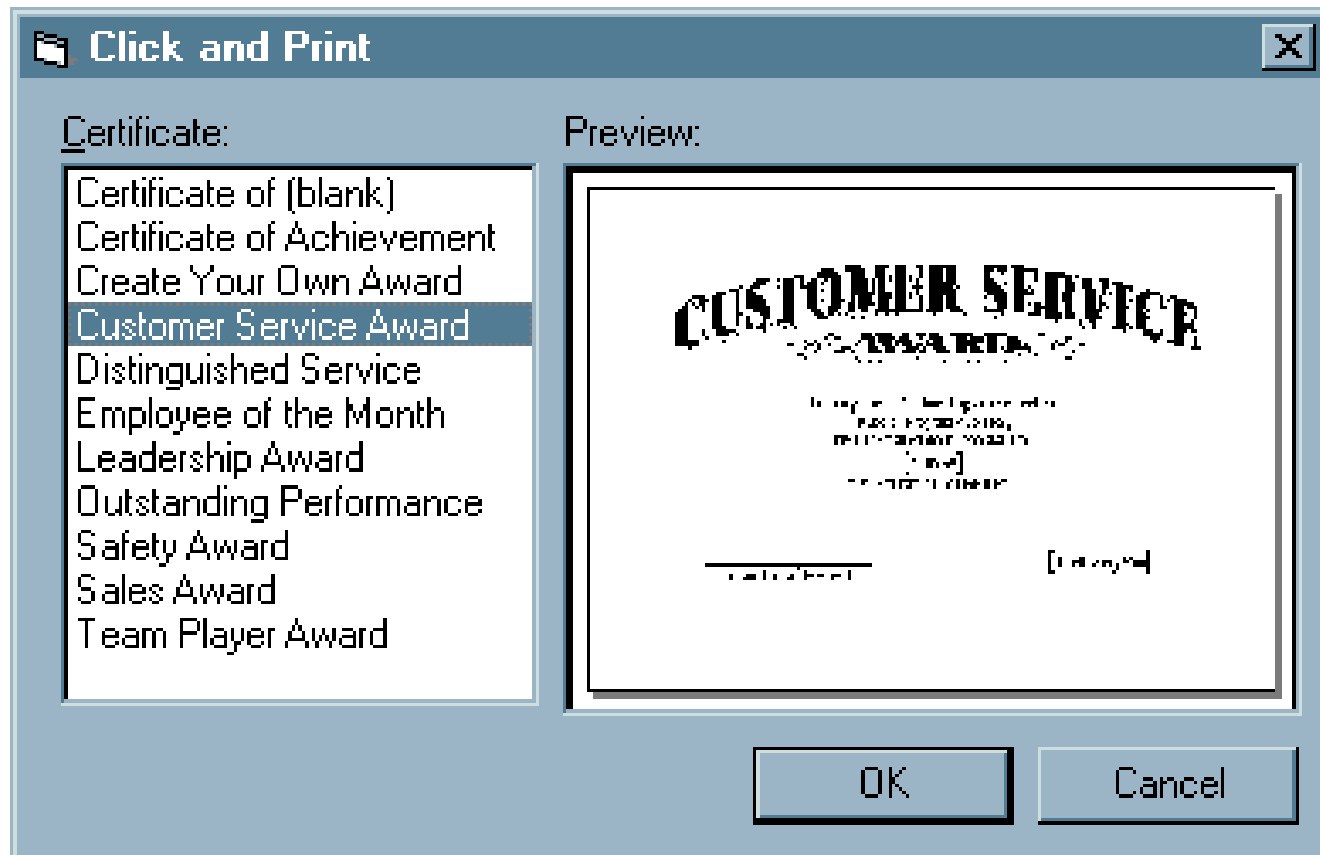
USABILITY

DINA UTAMI , M. SC

Interface Hall of Shame

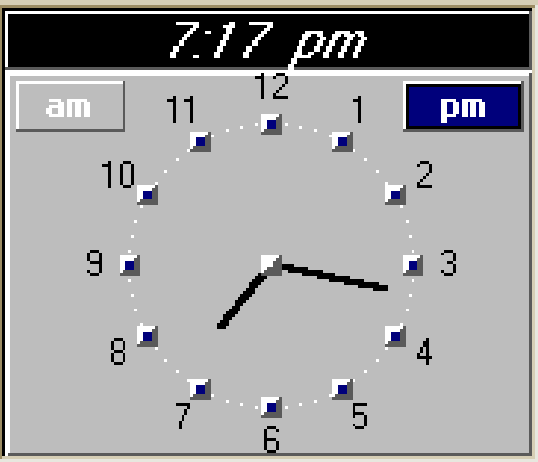


Desain yang Lebih Baik



Interface Hall of Shame

First Launch Date:	<input type="text" value="09/09/97"/>	<input type="button" value="Set Date"/>
First Launch Time:	<input type="text" value="19:17"/>	<input type="button" value="Set Time"/>



7:17 pm

am pm

Left Mouse Button: Change Minute
Right Mouse Button: Change Hour

OK Cancel

The image shows a classic analog clock interface for time selection. The clock face has numbers 1 through 12. The hour hand is positioned between 7 and 8, and the minute hand is positioned between 3 and 4. The time displayed is 7:17 pm. The interface includes 'am' and 'pm' buttons, a legend for mouse button actions, and 'OK' and 'Cancel' buttons at the bottom.

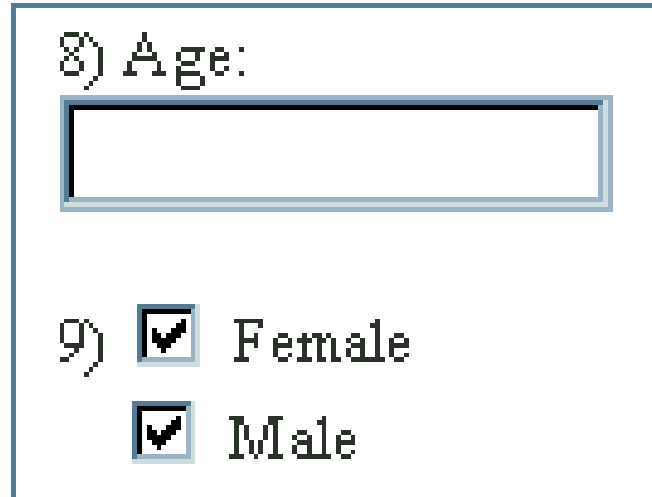
Interface Hall of Shame

8) Age:

9) Female
 Male

Apa yang salah?

Interface Hall of Shame



8) Age:

9) Female
 Male

Tidak MENCEGAH TERJADINYA EROR.

Interface Hall of Shame.

Model:

- Sport Coupe
- Sedan
- Sport Utility
- Pickup Truck

Apa yang salah?

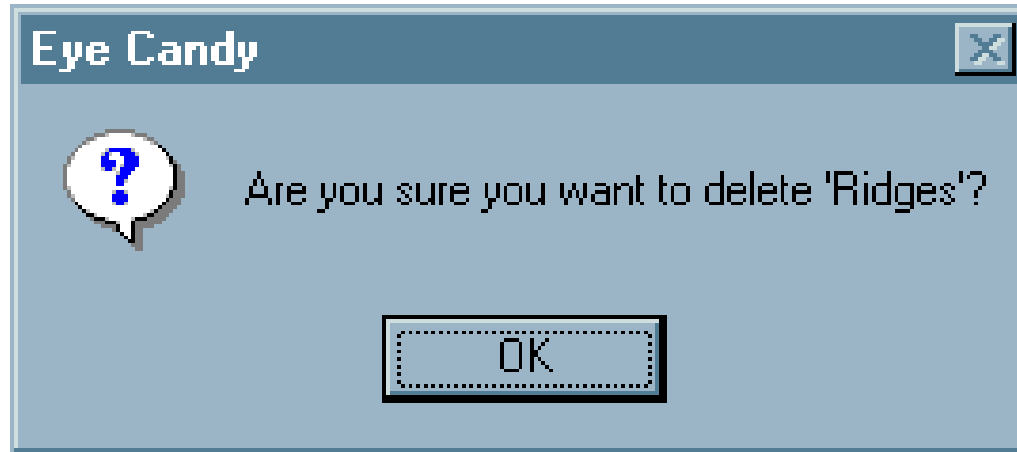
Interface Hall of Shame.

Model:

- Sport Coupe
- Sedan
- Sport Utility
- Pickup Truck

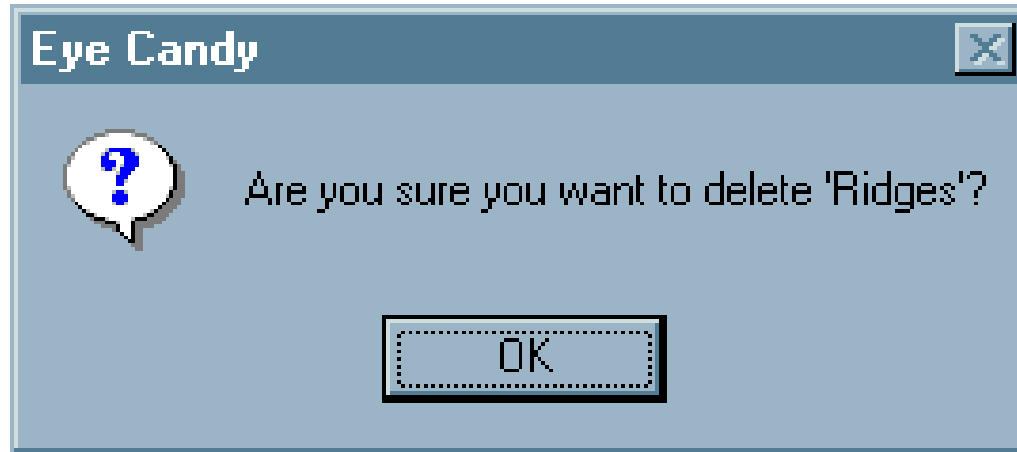
Tidak SESUAI DENGAN STANDAR.

Interface Hall of Shame.



Apa yang salah?

Interface Hall of Shame.



**Tidak MEMBERIKAN FASILITAS
UNTUK MEMBETULKAN KESALAHAN**

Mengapa Antarmuka Software Penting

- Antarmuka (*user interface*) sangat mempengaruhi persepsi atas sebuah software
 - Software yang '*usable*' penjualannya lebih baik
 - Website yang '*unusable*' ditinggalkan pengunjung

Konsekuensi Desain yang Buruk

- Software yang susah dipakai membuang-buang waktu user.
- Jika tidak dirancang dengan baik sekarang, maka harus dirancang ulang di kemudian hari.
 - Perancangan ulang membutuhkan biaya besar
- Dalam beberapa kasus memakan korban
 - Therac-25 radiation therapy machine
 - Aegis radar system in USS Vincennes

Sulitnya Mendesain Antarmuka

- Kamu bukan *user*-mu
 - UI membutuhkan banyak komunikasi dengan user.
- *User* selalu benar
 - Jika ada masalah dengan sistem yang terus menerus terjadi, berarti ada kesalahan dalam sistem.
- ...tapi ya tidak juga
 - User bukanlah desainer sistem

Sulitnya Mendesain Antarmuka

- UI membutuhkan usaha paling besar dalam pengembangan sebuah aplikasi.
- UI menghabiskan 50%:
 - Waktu untuk desain.
 - Waktu untuk implementasi
 - Waktu untuk *'maintenance'*
 - Ukuran *code* program

(survey of 74 software projects)

Pengertian Usability

- **Usability** -> Seberapa baik user bisa menggunakan fungsionalitas sistem.
- **Dimensi usability** (*harus bisa diukur*)
 - Learnability: Mudahkah dipelajari?
 - Efficiency: kalau sudah dipelajari, seberapa efisien bisa digunakan?
 - Memorability: Mudahkah mengingat yang sudah dipelajari?
 - Errors: Apakah banyak error dan apakah jika ada bisa diperbaiki?
 - Satisfaction: Menyenangkankah untuk digunakan?

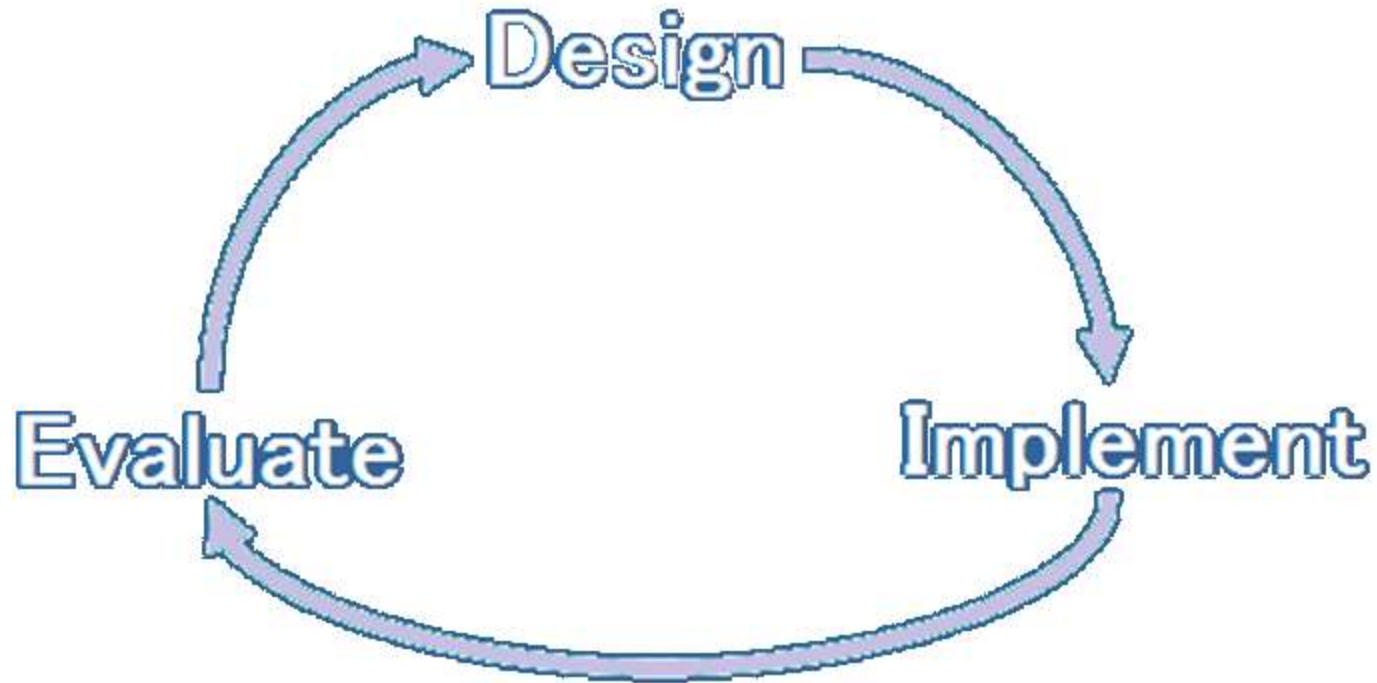
Tingkat Prioritas Dimensi Usability

- Berbeda-beda
- Tergantung user
 - ‘*Novice user*’ membutuhkan *learnability*.
 - User yang jarang menggunakan sistem membutuhkan *memorability*.
 - ‘*Expert user*’ membutuhkan *efficiency*.
- Tidak semua user selalu ‘*novice*’ atau ‘*expert*’
 - Pengalaman atas ‘*domain knowledge*’
 - Pengalaman menggunakan aplikasi
 - Pengalaman menggunakan fitur
 - » Contoh aplikasi stock market

Usability Bukan Satu-satunya

- Perancangan perangkat lunak harus memperhatikan hal-hal berikut:
 - *Functionality*
 - *Performance*
 - *Cost*
 - *Security*
 - *Usability*
 - *Size*
 - *Reliability*
 - *Standards*
- Banyak keputusan yang diambil saat desain yang mengharuskan tradeoffs beberapa atribut lain.
 - Contoh mode dalam besmart.
- Di kelas ini Usability nomor satu.

Usability Engineering: Sebuah Proses



Design

- Task analysis / Analisis Tugas
 - “Kenali User-mu”
 - *Siapa mereka? Apa yang sudah mereka tahu? Bagaimana lingkungan mereka? Apa tujuan mereka? Informasi apa yang mereka perlukan? Bagaimana mencapai tujuan?*
- Design guideline / Acuan desain
 - Menghindari kesalahan umum
 - Sifatnya tidak saklek
 - Terkadang kontradiktif dan ‘vague’
 - *User harus selalu memegang kendali, namun, diwaktu yang sama kita harus dapat mencegah error.*
 - *User harus selalu diberi tahu apa yang sedang terjadi, namun juga jangan lupa batasan mental workload mereka.*

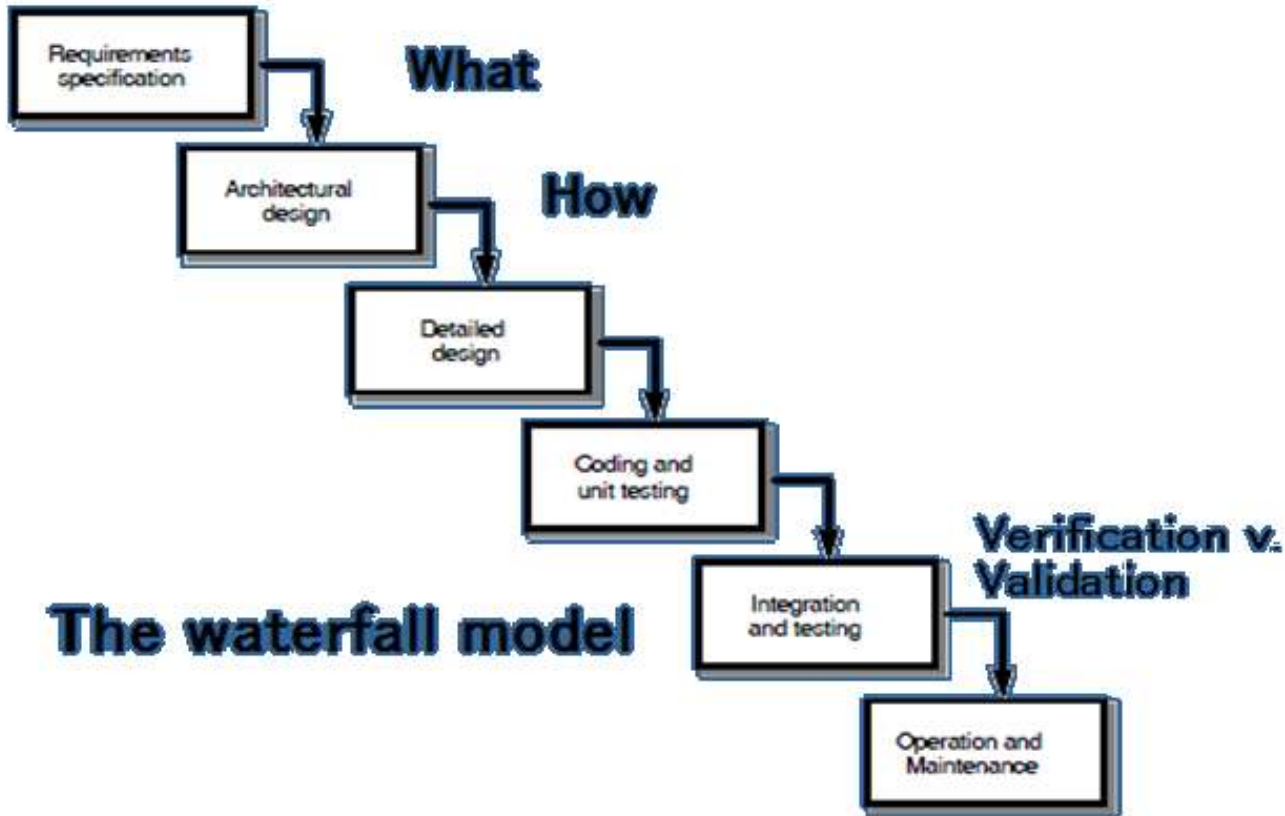
Implement

- Prototyping / Membuat Prototipe
 - Murah, Implementasi sekali pakai
 - Mudah dibuat: kertas, Wizard of Oz
 - Agak mudah dibuat: HTML, Visual Basic
- Teknik implementasi GUI
 - Model Input/output
 - Toolkits / alat bantu
 - UI builders

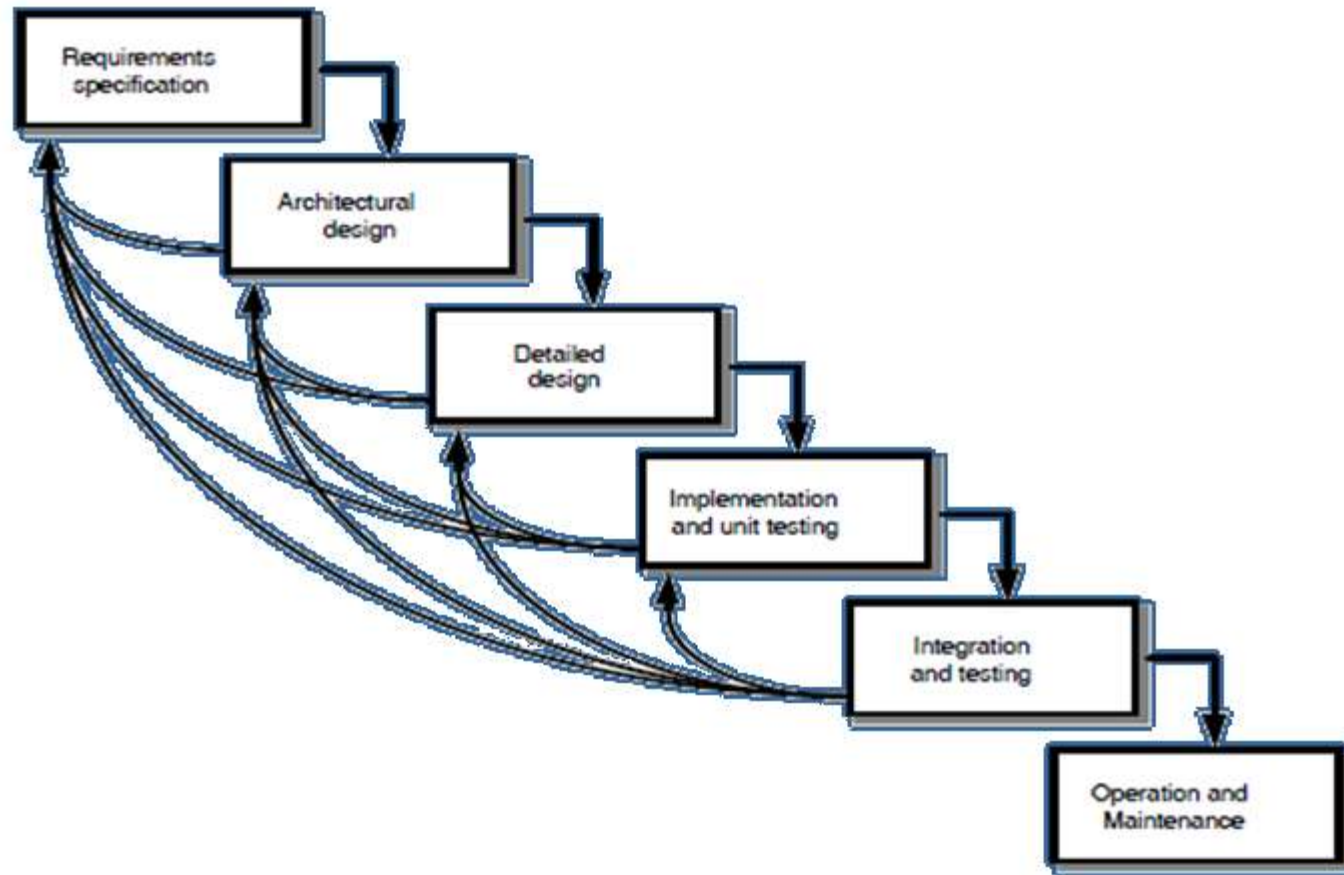
Evaluate

- Evaluasi bertujuan mengetes prototipe
 - Expert evaluation / Evaluasi oleh pakar
- Heuristics and walkthroughs
 - Coba-coba
- Predictive evaluation
 - Di-tes berdasarkan sebuah model (simulasi pengguna)
- Empirical evaluation
 - Melihat pengguna memakai

Software Lifecycle

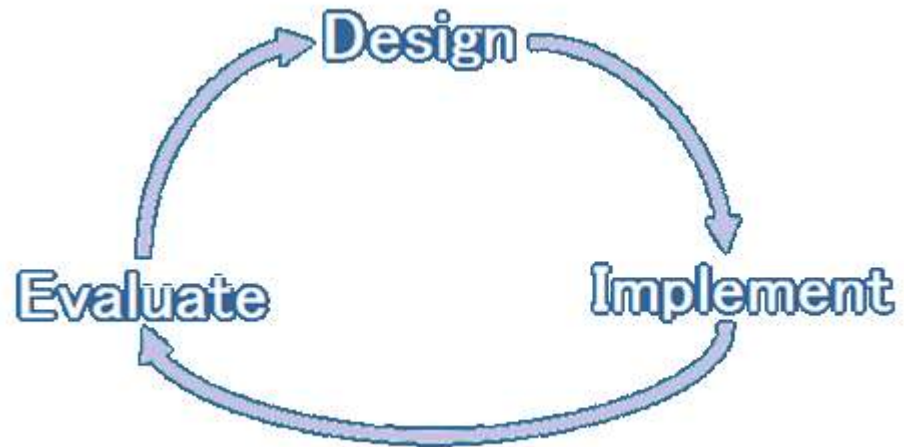


UI Lifecycle



Iterative Design

- Tidak bisa langsung benar
- Akan ada banyak siklus design-implement-evaluate.



Tujuan kuliah ini:

- In this course, you'll learn:
 - how to discover users' tasks
 - how human capabilities influence usability
 - guidelines for good UI design
 - the importance of iterative design
 - how to build cheap prototypes
 - techniques for UI implementation
 - how to evaluate UIs
 - Expert (heuristic) evaluation
 - User testing
 - current research in user interfaces