



الفرق بين التصميم والتحليل

د. محمد الأحمد

هندسة البرمجيات (1) 2018/12/22 RB Informatics ;

الفرق بين التصميم والتحليل

التحليل: الخدمات، المتطلبات الوظيفية وغير الوظيفية (فهم المشكلة)
التصميم: كيفية جلب نظام يحقق الخدمات ضمن القيود المفروضة
نعتمد على مخططات SRS بشكل أساسي التي هي دخل مرحلة التحليل وخرج مرحلة التصميم.
أول مرحلة بالتصميم هي التصميم المعماري (معمارية النظام) أي النظم الجزئية التي تكون النظام وكيف تتواصل هذه الأنظمة مع بعضها.

العوامل التي تؤثر على تحديد معمارية النظام

1. الوظائف المتشابهة تذهب لنفس النظام الجزئي.
2. التواصل بين النظم عن طريق المتطلبات الغير وظيفية بطريقة توزيع النظام وتكبير الأنظمة الجزئية أو تصغيرها مثلاً: إذا أردنا maintainability عالية (سهولة اكتشاف الأخطاء وتصحيحها) نقوم ب إكثار الأنظمة الجزئية فيكون لدينا Cabling عال مما يؤثر سلباً على الأداء (Performance) وبالعكس إذا أردنا security عالية نزيد الطبقات (layers) .
3. قيود الـ Hardware : القيود الفيزيائية مثل وجود قاعدة معطيات جاهزة.

أهم أنماط (المعماريات) الأساسية

1. MVC : بهدف عزل الـ Presentation عن طبقة الـ Data
 2. Client server : 2-tiers, n- tiers, fat client, thin client
 3. Repository : وتستخدم في ERP وهي packages software جاهزة لإدارة مؤسسات ضخمة
 4. Pipe and filter : تستخدم عندما دخل المرحلة هو خرج المرحلة التي تسبقها مثل الكومبايلر.
- يتألف التصميم من مرحلتين أساسيتين: التصميم المعماري والتصميم التفصيلي.

User Interface Design

إن خرج التصميم المعياري هو وثيقة تحوي الأنظمة الجزئية التي يتكون منها النظام وكيف تتفاعل مع بعضها، التصميم التفصيلي يعتمد على هذه الوثيقة كل نظام جزئي يجب تفصيله من حيث تصميم قاعدة المعطيات (تمت دراسته في مادة قواعد المعطيات) وتصميم الواجهات.

Graphic User Interface (GUI)

تصميم الواجهات الرسومية للمستخدم وتعد نقطة وصول الزبون للخدمات بقدر ما تصمم بشكل بسيط وسهل بقدر ما تعطي انطباع للزبون بأن النظام جيد وأحياناً يكون تصميم الواجهات الجيد يكافئ تطوير النظام كامل وقد يكون النظام سيء من حيث الوظائف لكن الواجهات جيدة فيقبل به الزبون لذلك فهي تحتاج عناية خاصة كون معيار النجاح هو رضا الزبون لذلك تكون الواجهات ممتازة بحسب طلب الزبون.

✓ تطوير الواجهات مرّ بمرحلتين أساسيتين:

○ واجهات كتابية: مثل نظام التشغيل Dos حيث كانت كل الخدمات كتابية عن طريق واجهة سوداء (تكمّن صعوبتها في حفظ التعليمات)

○ واجهات رسومية: بدأت بالظهور مع واجهات windows وهي أسهل للتعامل.

✓ وتنقسم الواجهات الرسومية إلى:

○ شخصية personal application (مثل word) الصفحة مكونة من قوائم، منطقة عرض، أزرار

○ Web application: صفحة الويب تعد بحد ذاتها واجهة.

هناك تشابه بينهما لكن في تطبيقات الويب نهتم بال navigation (التجوال) الموقع هو عبارة عن عدة صفحات (توجد صفحة رئيسية) يجب تنظيمها بطريقة علمية تفاعلية تقدم المعلومات بطريقة تزايدية حيث أن عدد التفرعات في الويب أكبر منها في التطبيقات الشخصية.

مثال: الصفحات التجارية (commercial) كل منتج يحتاج صفحة تحوي: ميزات - سعر المنتج - تاريخه - العروض..

حتى نتمكن من النجاح في تصميم الواجهات نحتاج في بعض الأحيان إلى:

○ مختصين ب Graphic designer

○ مختصين بالعلوم النفسية: مثلاً نريد تصميم لأشخاص مصابين بالتوحد فإننا بحاجة لمن يجيد التعامل معهم واختيار الألوان المناسبة.

ومنه يجب الاهتمام بـ:

○ **Layout تموضع العناصر:** يتم النظر للصفحة على أنها مجموعة أغراض (أزرار - منطقة عمل) وكيف يتم توزيع هذه الأغراض.

○ **الألوان (اختيار الألوان المناسبة للفكرة):** مثال تطبيق ويب لدفن الموتى لا يناسبه الألوان الزاهية وإنما ألوان عاتمة ويجب أيضاً احترام ثقافة الشعوب والأديان والرموز.

○ **سهولة التعلم والاستخدام والفهم:** نلاحظ وجود فرق بين Web application و Personal Application

أن تطبيقات الويب لا تحوي دليل استخدام (manual) أثناء التصميم يجب أن نساعد المستخدم أن يعرف مكان وجوده ونسهل عملية التصفح عن طريق مجموعة قواعد تسمى قواعد الانسجام والاتساق والتي نراعيها أثناء التصميم فالتطبيق يجب أن يصل المستخدم فيه إلى الخدمات دون الحاجة إلى إرشادات وإنما بال hint والرموز فقط، بينما لو كان لدينا تطبيق للجامعة نقوم بتعليم الموظفين كيفية استخدام كل واجهة عن طريق دليل

استخدام كما يجب عدم وضع مفاهيم جديدة للرموز المعروفة مثل زر الاغلاق يعبر عن ب × ، رمز

الdownload.....

بعض الأخطاء (المعيارية في تصميم الواجهات):

1. نقص الاتساق والانسجام: إعطاء معانٍ جديدة لرموز معروفة.
2. تحتاج إلى تذكّر الكثير لتصل إلى المعلومات والخدمات.
3. عدم وجود إرشادات (لا تستطيع الوصول إلى الخدمة).
4. لا يوجد حساسية للسياق: مثلاً في الصفحة الرئيسية توجد الخدمات العامة وكلما انتقلنا لصفحات كلما انتقلنا لخدمات خاصة أكثر.
5. استجابة ضعيفة (مدة طويلة في تحميل الخدمات ...): غالباً الصفحة تعمل على شكل threads وقد تحتوي على محتويات مختلفة (صور، نص، فيديو....) تختلف في مدة تحميلها فيجب مراعاة ذلك أثناء التصميم مثلاً google chrome يوفر للمستخدم ميزة تخطي الصور لتحميل أسرع أثناء تصفح البريد الإلكتروني.
6. Unfriendly: غير مريح للعين، المحتوى غير متناسق، لا يوجد تناسق في Layout.

القواعد الذهبية أثناء التصميم

1. وضع المستخدم في موضع التحكم باستخدام الصفحة: في أي لحظة يعلم مكان وجوده في أية صفحة، الخروج في الوقت الذي يريد، العودة للخلف في أي وقت
2. التقليل من استخدام ذاكرة المستخدم.
3. الاتساق والانسجام.

وضع المستخدم في موضع التحكم باستخدام الصفحة

1. تحديد أوضاع التفاعل بطريقة لا تجبر المستخدم على اتخاذ إجراءات غير ضرورية أو غير مرغوب فيها ...
2. توفير مرونة بالتفاعل.
3. السماح بتفاعل المستخدم بحيث يكون قابلاً للانقطاع وغير قابل للإزالة (نوع من التحكم مثل office ووجود خدمة التراجع عن بعض الخدمات).
4. تبسيط التفاعل مع تقدم مستويات المهارة والسماح بتشخيص التفاعل (بعض الصفحات تسمح للمستخدم بتغيير الألوان) التركيز على ما يفضلها المستخدم.
5. إخفاء التقنية الداخلية عن المستخدم العادي فالتطبيق ليس فقط للأشخاص المتعلمين وإنما لكل الفئات ويمكن أخذ عينات مستخدمين لقياس سهولة الاستخدام ومراعاة كافة شرائح المستخدمين مما يوفر Usability للنظام، أي توفير واجهات مختلفة بتصاميم مختلفة لأشخاص مختلفين حسب خلفياتهم الثقافية والاجتماعية.
6. تصميم للتفاعل المباشر مع الكائنات، أي جعل عملية التواصل مباشرة وبنقرة أو ...

(التقليل من استخدام ذاكرة المستخدم)

للصور أنماط مختلفة كل نمط يأخذ حجم مختلف ويجب أخذ ذلك بعين الاعتبار فالصور ذات الحجم الصغير يكون تحميلها أسرع واستجابة الواجهة للصور تكون أسرع لدى المستخدم فمثلاً لا نأخذ صيغة الصورة jpeg وإنما نستخدم png نفس الصورة تأخذ حجم أقل وهو موضوع تقني وصحيح أن الحجم يقل لكن الوضوح يقل أيضاً لذا يجب مراعاة أن يكون الصورة واضحة مع تخزين أقل.

الاتساق والانسجام

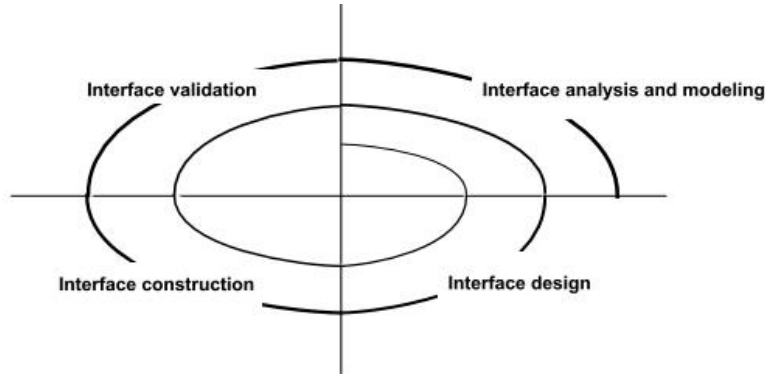
- عدم اعطاء معانٍ جديدة للمفاهيم القديمة مثل × للإغلاق.
- الحفاظ على التناسق عبر مجموعة من التطبيقات: عندما يتعلم المستخدم استخدام واجهة تطبيق معين يصبح قادراً على استخدام أي من التطبيقات الأخرى.
- إذا كان النماذج التفاعلية السابقة قد خلقت توقعات للمستخدمين وعدم خرق القواعد العامة المتعارف عليها، فلا نقوم بتغيير ما لم يكن هناك سبب مقنع للقيام بذلك.

خطوات تصميم واجهة

هي عملية تكرارية لا تتم من أول مرة، نبدأ من مخطط ال Activity Diagram لنقوم بتعريف الواجهات الأساسية التي يجب أن يوفرها للمستخدم، من خلال فهم المخططات نستخلص الواجهات التي يجب أن تكون موجودة، مثلاً: في حال وجود عملية تسجيل دخول يجب تصميم الواجهة المناسبة مع مراعاة سيناريوهات النجاح والفشل وتوفير رسائل الخطأ والتحذير.

ننتقل من User Model: هو ملف تعريف لجميع المستخدمين النهائيين للنظام فيجب مراعاة المستخدمين وثقافتهم ليتم تصميم واجهات ملائمة لهم.

يجب وضع Design Model: تحقيق تصميم النموذج عبر تحويل التصميم إلى كود، عندما نصمم تطبيقات شخصية نستخدم الأدوات الخاصة من واجهات جافا (swing) أو c# أما تطبيقات الويب نستخدم HTML & CSS.



وكما ذكرنا فهي إجرائية تكرارية نبدأ من التحليل بعملية Interface Analysis and Modeling ثم وضع التصميم المناسب وتنفيذه بإحدى اللغات حسب نوع التطبيق وأخيراً عملية Validation من قبل الزبون للحصول على واجهات معيارية.

Interface Analysis

تحليل الواجهة يتم من خلال:

1. فهم الأشخاص (المستخدمين النهائيين) الذين سيتفاعلون مع النظام من خلال الواجهة ومعرفة توجهاتهم وما هو شكل الواجهة والخدمات التي تقدمها.
2. فهم المهام التي تقدمها الواجهة: لا يجب دائماً بناء واجهة لكل خدمة بل يتم تجميع عدة خدمات متشابهة في واجهة واحدة وترتيبها ضمن menu لكن بعض الخدمات تحتاج واجهة لوحدها أما الخدمات المترابطة التي تحتاج شروط مسبقة واحدة تكون في واجهة واحدة.

3. المحتوى الذي يتم تقديمه كجزء من الواجهة من نصوص وصور وفيديوهات هذا المحتوى يجب أن يكون فعال ومترابط ومناسب لحالة الواجهة.
4. البيئة التي سيتم فيها هذه المهام.

لدينا مجموعة أسئلة نعلم من خلالها كيفية تصميم الواجهات والخدمات التي تتضمنها:

غالباً يكون الزبون غير قادر على تحديد الخدمات فنلجأ إلى طريقة Prototyping وهي مجموعة نسخ تجريبية ورقية فنرسم التطبيق كامل بدءاً من الواجهة الرئيسية ومروراً بجميع الواجهات ثم يجري التناقش مع الزبون لإجراء التعديلات المناسبة ثم ننتقل لتنفيذ التصميم برمجياً وهناك مجموعة أسئلة تساعدني في هذه المرحلة.

Task Analysis and Modeling

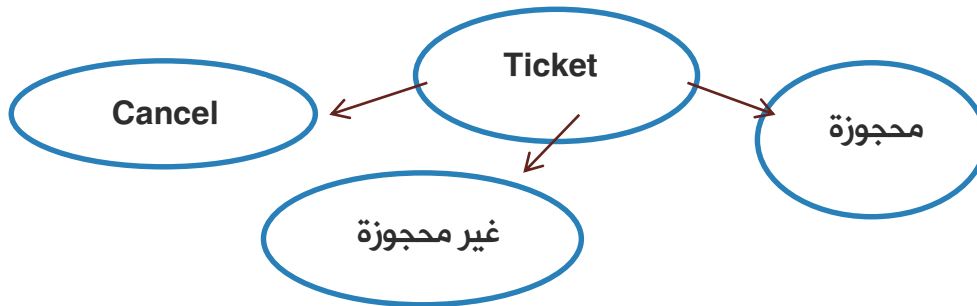
بعد فهم الخدمات يوجد خطوات مساعدة لذلك:

(1) Use Case: الهدف منه توضيح التفاعل بين النظام والوسط الخارجي (actors) ومنه نبدأ منه لأن الوصول للخدمات يتم عبر الواجهات فمن فهمنا له يمكننا وضع التصميم.

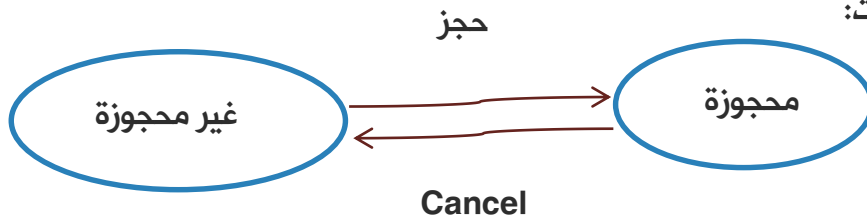
(2) Activity Diagram: يركز على السيناريوهات المحتملة ومنه يعطينا نظرة متوسطة التجريد للنظام (Middle Abstract) فيعرض النظام بشكل مفصل لكن من وجهة نظر خارجية كتسلسل خدمات أي تجريد متوسط للنظام وكيف تتم الخدمات دون التطرق لبنية النظام ومنه فإنه يمكننا بناء واجهات لسيناريوهات النجاح والفشل

State Diagram

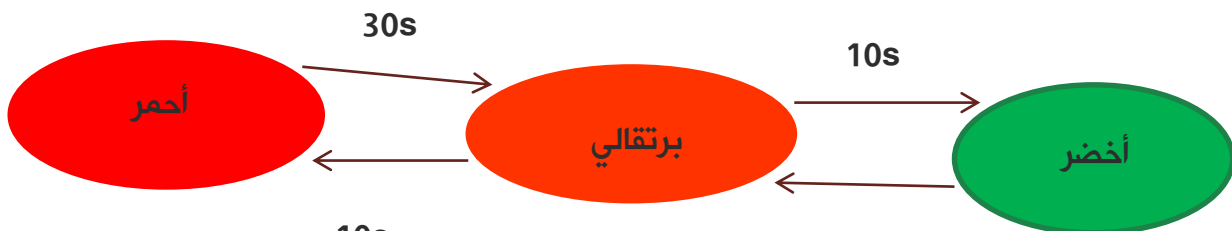
"تابع للمخططات" هو مخطط غير متعلق بـ Use Case يدرس حالة غرض معين من صف ما عبر استخدامه في النظام مثال:



والانتقالات بين هذه الحالات:



مثال:



يتم بعد رسم مخطط الصفوف وليس 30s كل مخططات الصفوف ولكن للم 10s .

Analysis of Display Content

معرفة المحتوى المناسب من نص وصور أيضاً اللغة المناسبة والتي تختلف من مكان لآخر حسب الثقافة اعتماداً على المستخدمين النهائيين.

خطوات تصميم الواجهة

بعد تحديد المستخدمين والخدمات والمحتوى الذي نريد تقديمه نبدأ ب Prototyping ويتم بشكل تكراري بإضافة تعديلات وتحسينات بشكل حلزوني.

بعض القضايا التصميمية

1. Response time:

الاهتمام بزمان الاستجابة عبر معرفة الخلفيات التقنية من التحميل والتحليل و...

2. Help facilities:

تأمين التسهيلات المساعدة للمستخدم لتسهيل الوصول إلى الخدمات ومعرفة مكان التواجد عبر أسهم ودلالات توضيحية.

3. Error handling:

التعامل مع الأخطاء مثل أخطاء بالإدخال فيجب معالجتها وتنبيه المستخدم.

4. Menu and command labeling:

وجود قوائم وسطور أمر.

5. Application accessibility:

امكانية الوصول إلى التطبيق.

6. Internationalization:

توافر عدة نسخ للتطبيق بعدة لغات.

WebApp Interface Design

❖ صفحة html هي الواجهة بحد ذاتها، وتتكون بشكل رئيسي من:

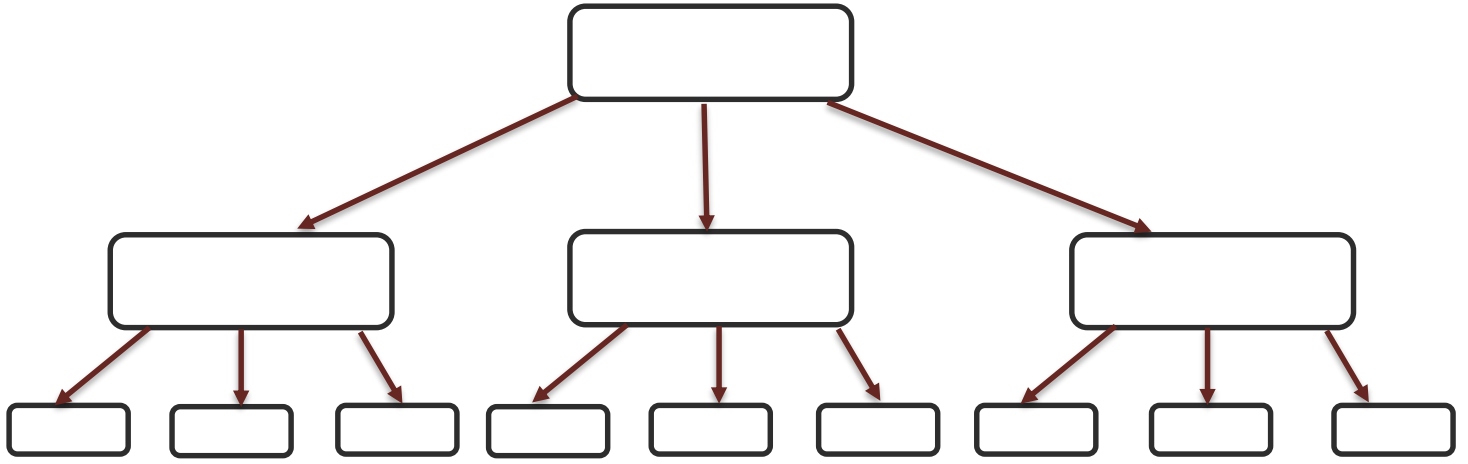
1. الصفحة الرئيسية: تحتوي لوغو الشركة أو المؤسسة، ويكون موجود في جميع الصفحات أيضاً، فيها تعداد لكل الخدمات العامة وعن طريقها يمكن الوصول إلى الصفحات الداخلية.

2. الصفحات الداخلية.

3. التجوالية: طريقة التجول بين الصفحات وربط الصفحات.

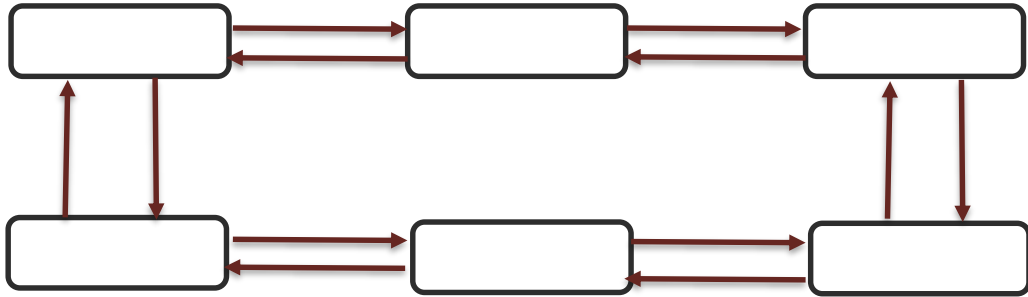
❖ طرق ربط الصفحات:

1. أفضل طريقة هي الطريقة الهرمية



2. الطريقة الشبكية: الوصول من كل صفحة إلى كل الصفحات.

3. الطريقة المصفوفة:

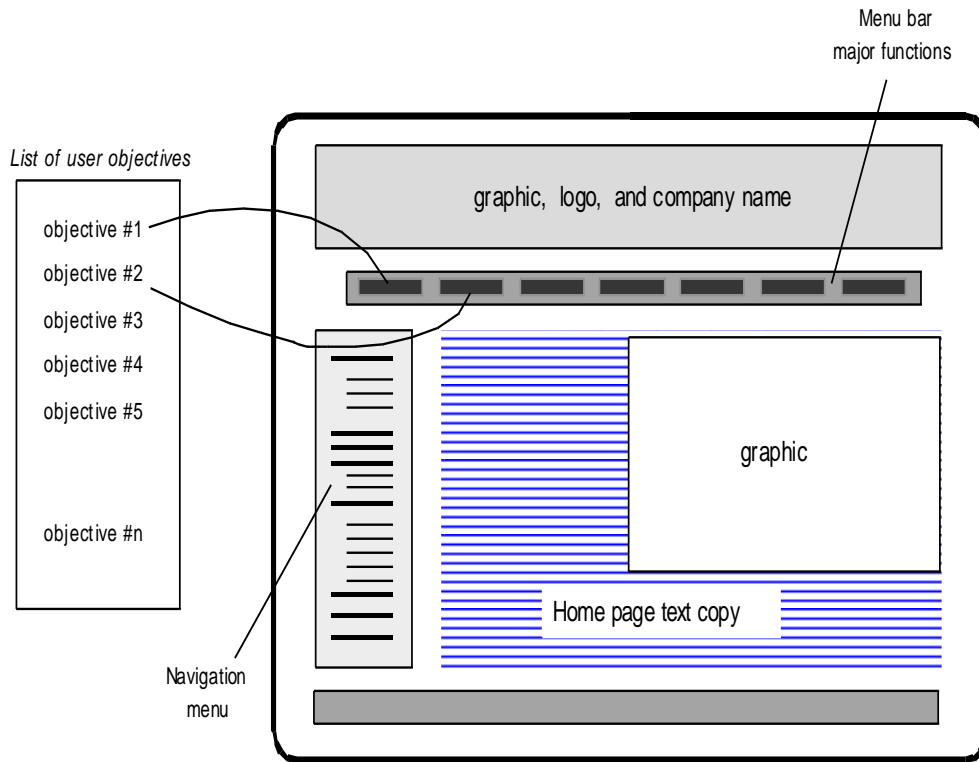


تستخدم في الصفحات التجارية (موقع البيع) وهي غير مريحة بالنسبة للمستخدم costumer ، لكن تستخدم لعرض منتجات أكثر حيث يضطر المستخدم إلى المرور في تلك الصفحات.

Interface Design Workflow

- **Anticipation التوقع:** يجب تصميم webApp بحيث يتوقع حركة المستخدم التالية.
- **Communication الاتصال:** يجب أن تقدم الواجهة بإبلاغ الحالة بأي نشاط يقوم به المستخدم.
- **Consistency الاتساق:** استخدام عناصر التحكم، القوائم، الرموز وعلم الجمال (مثل اللون والشكل والتخطيط).
- **Controlled autonomy الاستقلالية:** يجب أن تسهل الواجهة حركة المستخدم عبر webapp
- **Efficiency الكفاءة:** تصميم الWebapp والواجهة يجب أن يعملان على تحسين كفاءة عمل المستخدم وليس كفاءة المهندس الذي قام بتصميمها وبناءها أو بيئة خادم العميل الذي يقوم بتنفيذها.
- **Focus التركيز:** يجب أن تظل واجهة webapp والمحتوى الذي يقدمه مركز على مهمة المستخدم الذي يقوم به.
- **Fit's Low:** إن الوقت للحصول على الهدف هو وظيفة المسافة إلى حجم الهدف.
- **Human interface objects البشرية:** تم تطوير مكتبة ضخمة قابلة لإعادة الاستخدام ل webapp.

- **تقليل وقت الاستجابة Latency reduction:** webapp يجب أن يستخدم تعدد المهام بطريقة تسمع للمستخدم بمتابعة العمل كما لو كانت العملية قد اكتملت.
- **القدرة على التعلم learn ability:** واجهة الwebapp يجب أن تصمم لتقليل وقت التعلم وبعد تعلمها، لتقليل إعادة التعلم المطلوبة عند إعادة النظر في ال webapp .
- **الحفاظ على سلامة منتج العمل:** على سبيل المثال: نموذج مكتمل من قبل المستخدم قائمة محددة من المستخدمين، يجب حفظ منتج العمل تلقائياً حتى لا يتم فقدته في حالة حدوث خطأ.
- **قابلية القراءة:** يجب أن تكون كل المعلومات المقدمة عبر الواجهة قابلة للقراءة من قبل الصغار والكبار.
- **حالة التتبع:** عند الحاجة، يجب تتبع حالة تفاعل المستخدم وتخزينها بحيث يتمكن المستخدم من تسجيل الخروج والعودة لاحقاً إلى المكان الذي توقف عنده.
- **النقل المرئي:** واجهة الWebApp المهمة جيداً، تؤمن خيال أن المستخدمين في نفس المكان مع جلب العمل لهم.
مثال:

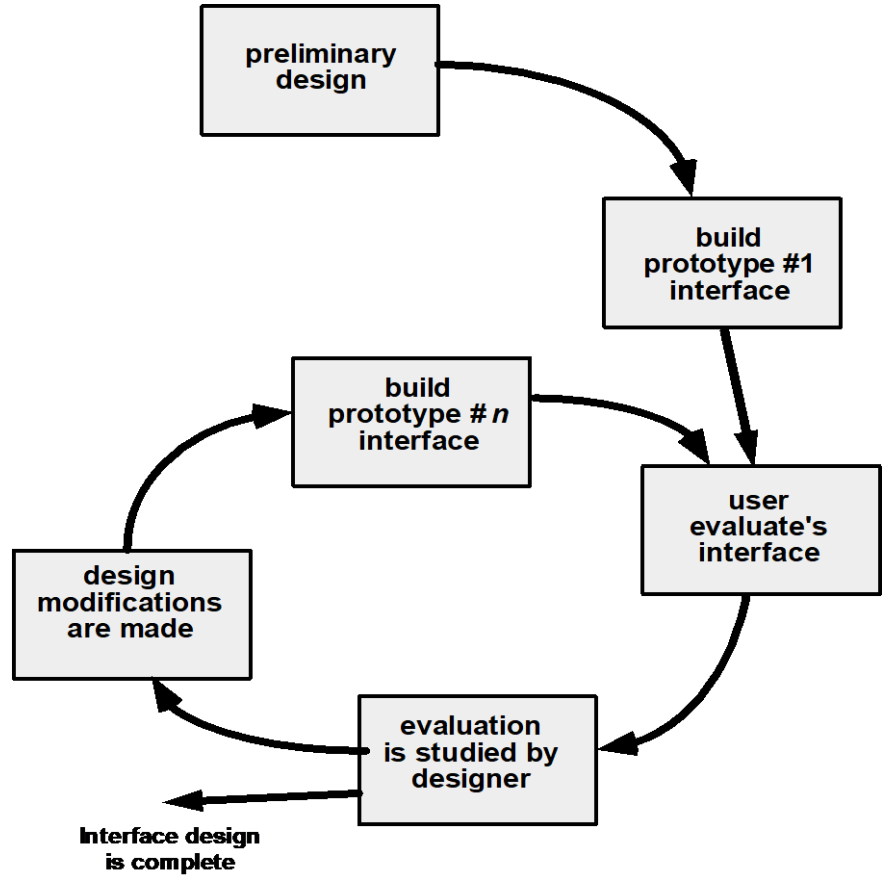


Aesthetic Design

- لا تخف من المساحات البيضاء: العين ترتاح لرؤية اللون الأبيض فيجب الابتعاد عن حشو الكثير من المحتوى في الصفحة.
- التأكيد على المحتوى: فلو كانت الخدمة رسالة يجب التركيز عليها وعدم وضع رسائل إضافية بدون فائدة.
- تنظيم عناصر التخطيط من أعلى اليسار إلى أسفل اليمين
- ترتيب عملية التجوالية على شكل مجموعات.

- محاولة عدم استخدام Scroll Bar واستخدام الشريط العمودي بدل الأفقي عند الحاجة.
- الاهتمام بالدقة وحجم النافذة والصور عند التصميم.

Design Evaluation Cycle



انتهت المحاضرة