

Université : Batna 2
Faculté : mathématique et informatique
Département : informatique

Chapitre 6

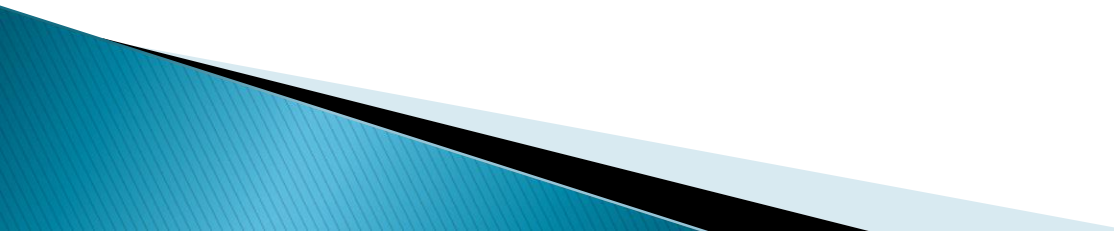
Application spécifique & Conception d'interface graphique

Application spécifique

Introduction

- ▶ En informatique, un **logiciel spécifique** est un logiciel développé sur commande à l'attention d'un client donné.
- ▶ Un logiciel standard, qui est un logiciel développé sur initiative d'un éditeur, et vendu à de nombreux clients.
- ▶ Le terme anglais correspondant à logiciel spécifique est "*custom software*", ou "*bespoke software*".
- ▶ Les Britanniques parlent de *bespoke development* pour désigner le développement spécifique (développement d'un logiciel spécifique).

Introduction

- ▶ La construction d'un logiciel spécifique est une prestation de service, qui consiste à fournir l'**expertise technique et la main d'œuvre nécessaire**.
 - ▶ Les fonctionnalités, le planning de livraison, et les conditions de paiement font l'objet d'un **contrat** entre le prestataire et le client.
 - ▶ Le consommateur est fortement impliqué dans le processus de construction et signe la réussite du travail.
- 

Définition

- ▶ Les logiciels spécifiques sont des produits créés **sur commande**, quand il n'existe aucun logiciel standard équivalent, dans des domaines très spécialisés
- ▶ La quasi totalité des logiciels spécifiques sont des logiciels applicatifs, et il n'existe pratiquement aucun logiciel système ni jeu vidéo spécifique
- ▶ Les logiciels standards répondent souvent à un nombre limité, voire insuffisant de variations, et c'est ce créneau qui est exploité par les logiciels spécifiques

Développement

- ▶ La construction d'un logiciel spécifique est une prestation de service,
- ▶ une production à l'unité et sur commande d'un logiciel unique
- ▶ La construction fait l'objet d'un contrat individuel entre le développeur et l'acheteur, Le contrat fixe
 - Les fonctionnalités attendues du logiciel
 - Le planning de livraison et de paiement

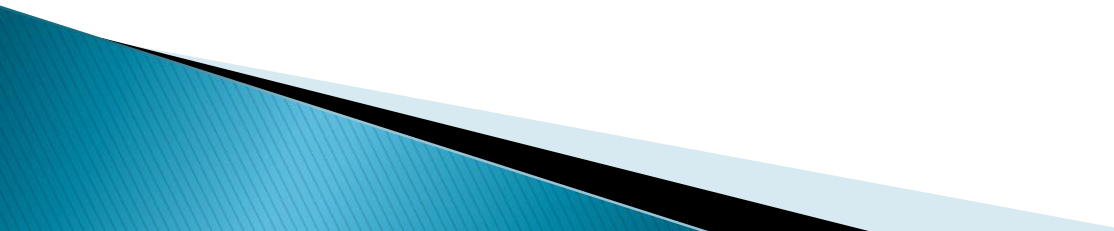
Les fonctionnalités

- ▶ La production du logiciel se fait graduellement en plusieurs phases.
- ▶ à la fin de chaque phase le client reçoit une version du logiciel
- ▶ Chaque phase se termine par une étape de test d'acceptance, par laquelle le client vérifie que le logiciel fait bien ce qui est attendu
- ▶ Le logiciel est alors testé dans de nombreuses conditions, avec des données réelles, éventuellement accompagnée de tests de stress destinés à faire échouer le logiciel et voir comment il récupère

Le planning de livraison

- ▶ Le planning en plusieurs phases permet de tenir compte de l'évolution des attentes du client: la conclusion d'une phase peut inspirer le client, l'amener à demander un produit plus raffiné et il est très rare que le produit final soit ce qui avait été envisagé au départ

Le paiement

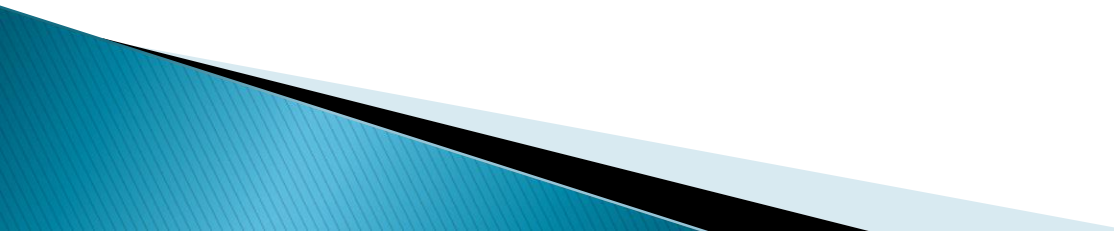
- ▶ Le paiement peut se faire sur base horaire - paiement régulier des heures de travail du développeur, ou au forfait - un prix fixe est négocié à la conclusion
 - ▶ Lors d'un paiement au forfait, l'évolution des demandes du client peut entraîner une renégociation du contrat, et une modification du prix
- 

Conception graphique de l'interface

Introduction

- ▶ Dans le design, il existe un certain nombre de règles à suivre. Y compris quand il s'agit de créer une interface, qui doit être à la fois
 - pratique,
 - Utile,
 - Et agréable à utiliser.
- ▶ La règle des trois C peut vous aider à vérifier si vous respectez ces objectifs. Les 3 C, ce sont les couleurs, les contrastes et le contenu

Premier C : les couleurs

- ▶ Les couleurs ont impact au niveau psychologique
 - ▶ les couleurs influencent non seulement la vision que les clients potentiels auront de vous, mais joue aussi sur leurs intentions d'achat)
 - ▶ Dans le cadre du design d'interface, les teintes retenues devront également servir à guider l'utilisateur, à transformer son expérience
- 

Deuxième C : les contrastes

- ▶ Le contraste, quand on parle de design d'interface, ne se limite pas à vérifier l'impact d'une page quand elle est imprimée en noir et blanc.
- ▶ Le contraste va créer du dynamisme en jouant sur :
 - Les couleurs
 - Les tailles des éléments
 - Leurs formes
 - Leurs directions
- ▶ En alternant les formes et les tonalités claires ou foncées, vous devez uniquement chercher à aider l'internaute à mieux se repérer : faire joli n'est pas la priorité aux premiers stades du développement.

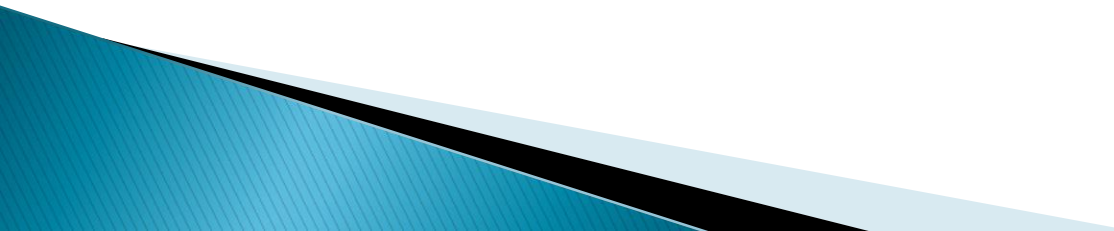
Troisième C : les contenus

- ▶ Sur une interface, vous trouverez principalement trois types de contenus. Chacun d'entre eux doit être utilisé uniquement s'il est intéressant pour l'utilisateur.
 - Des images
 - Du texte
 - Des vidéo

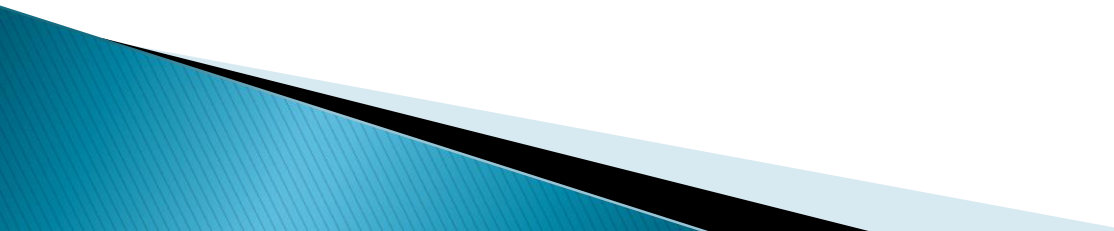
Des images

- ▶ Les images peuvent avoir deux fonctions sur une interface : être utiles ou créer un lien émotionnel
- ▶ Parmi elles, vous trouverez
 - **Des photos** (de produits, de démonstration ou d'appui par rapport à un contenu textuel),
 - **Des illustrations** (qui personnalisent l'interface),
 - **Des icônes** (au but principalement utilitaire, mais qui peuvent être transformées pour leur insuffler de la personnalité),
 - **Des mascottes** (pour créer un sentiment d'appartenance).

Du texte

- ▶ Le texte ne fait pas partie du design proprement parler, mais il est pourtant un élément indispensable pour toute interface, il faut donc lui trouver de la place et le mettre en valeur.
 - ▶ La typographie, la taille, la couleur du contenu textuel doit donc également être pensée et prévue pour attirer l'œil et pour être lisible par tous.
- 

Des vidéos

- ▶ Les vidéos sont maintenant devenues incontournables et on en trouve également dans les interfaces, que ce soit en guise d'introduction, de démonstration ou juste pour illustrer un propos.
 - ▶ Dans ce contexte, cependant, les vidéos devront être prévues pour correspondre au ton de l'interface et au reste de son contenu, pour créer un effet d'unité réel.
- 

La règle primordiale

- ▶ Pensez toujours d'abord à l'utilisateur.

