

ESIGELEC

Cours UML

Février 2007

Partie 1

Plan

• Aujourd'hui

- Ce qu'est UML
- Programme de mise à niveau

Plan

• Aujourd'hui

- Présentation (20 min)
- Construire du logiciel (60 min)
- Logiciel O.O. = modélisation sociale (20 min)
- Le paradigme O.O. (60 min)

Plan

• Présentation

- Vous
- Moi
- Votre (futur) métier
- Notre relation
- T.P.
- Evaluation de votre travail
- Support en ligne

Plan

• Construire du logiciel

- Les métiers du soft
- Votre expérience du dev. en équipe
- La gest. de projet
- L'estimation du projet
- Les tests
- Le génie logiciel
- PAQ & Plan de Configuration



Plan

• Logiciel O.O. = modélisation sociale

- Ex : Modélisation par exemple
 - Sociologie des organisations appliquée à la création d'une entreprise



Plan

• Le paradigme O.O

-



Présentation>vous

• Vous

Présentation>moi

Présentation

Présentation>moi

Actuellement

- Consultant en systèmes d'information
 - Pour des Start-up et sociétés orientés high-tech
- Etudes et Développements
 - Orienté Web, marketing web, eCommerce, stratégie commerciale et animation de communauté...

Présentation>moi

Avant

- Master MPI au CESI de Rouen
- Consultant eBusiness
 - Royal Orchid
- Responsable Informatique
 - Ecole d'Architecture de Normandie
- C.N.R.S.
 - Thèse pro sur la simulation logistique et les SMA

Présentation>moi

DscBox

- Système de portail captif
- Clients finaux : Pasino Partouche, hôtels...
- Missions, développement (à partir du logiciel existant) :
 - d'un système de gestion des utilisateurs
 - d'un système de gestion des log de connexions
- Compétences : PHP5, MySQL, Linux Debian, iptable, UML, reverse ingeneering...

Présentation>moi

BlogPerformance (pour Born4Sport)

- Intégration d'un Blog à un système de SingleSignOn développé en ASP .net par le biais d'un Webservice
- Développement en PHP 5, reserve ingeneering, WebServices
- Développement spécifique sur des systèmes de blog, installation et sécurisation de serveurs Linux...

Présentation>moi

- Audit de l'ensemble du cycle de vie d'un logiciel réalisé par une équipe de 9 ingénieurs
- Réalisation de documentation technique et présentation devant le responsable national du client final (taille de l'entreprise > 50000 collaborateurs)
- Aide à l'acquisition d'ouvrage
- Conseil (Informatique nomade, wifi, urbanisation des S.I. ...)

Présentation>moi

Institut ESAS

- Membre et conférencier pour l'Institut Européen des Sciences Avancées de la Sécurité

Présentation>moi

Linutop



- Le Linutop est un PC de la taille d'une cassette vidéo
- Conseil en stratégie et création de CaseStudy
- Conseil technique
- Création de maquettes logicielles (distro Linux)
- ...

Présentation>moi

nomade21.tv

- Lancement d'une WebTV en collaboration avec la maison d'édition M2Editions

M2Editions & Eyrolles Editions

- Relecteur technique

Présentation>moi

Partenaire :

- Nokia 770 developer
- Nokia 7710 VIP
- Palm Specialist
- ...

Présentation>moi

Contact

- Site Web/Blog/Wiki... : www.desechalliers.com
- Email : laurent@desechalliers.com
- Skype : desechalliers
- Tel : 06.24.79.15.95

Présentation

• Votre (futur) métier

- Pouvez-vous définir votre futur métier ?

Présentation

• Notre relation

- Evaluation uniquement sur vos résultats
 - Et non sur votre présence
- Si vous venez en cours, c'est pour participer
- N'hésitez pas à m'interrompre
 - m'interroger
 - ...

Présentation

• T.P.

- Basés sur des cas d'étude (simplifiés)
- Choix d'énoncés « flous »
 - Car reflet de la « réalité client »
- Suis à votre disposition mais vous devez tendre à devenir autonome

Présentation

• Evaluation de votre travail

- Evaluation uniquement sur la qualité du travail final
 - Pas d'évaluation morale (sur la présence)
- Examen : un cas d'étude
 - Pas de QCM
 - Droit à tous les supports papier
 - Mais pas électronique

Présentation

• Support en ligne

- Diaporama des cours seront disponibles en ligne
 - Formats : Open Document, PDF
- Librement diffusables et modifiables

**Documents sous Creative Common
by-nc-sa 2.0**

Présentation

Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public
- de modifier cette création



Selon les conditions suivantes :



Paternité. Vous devez citer le nom de l'auteur original.



Pas d'Utilisation Commerciale. Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.



Partage des Conditions Initiales à l'Identique. Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

- A chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création.
- Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.

Construire du logiciel

- Les métiers du soft
- Votre expérience du dev. en équipe
- La gestion de projet
- L'estimation du projet
- Les tests
- Le génie logiciel
- PAQ & Plan de Configuration

Construire du logiciel

- **Les métiers du soft**
 - Quels sont les métiers du logiciel ?

Construire du logiciel

- **Votre expérience du dev. en équipe**
 - Quelle est votre expérience du dev. en équipe ?
 - Quel sont les outils indispensables au dev. en équipe ?

Construire du logiciel

- **La gest. de projet**
 - FAISABILITE
 - ETUDE PREALABLE
 - SPÉCIFICATIONS GÉNÉRALES
 - SPÉCIFICATIONS DÉTAILLÉES
 - CONCEPTION GÉNÉRALE
 - CONCEPTION DETAILLÉE
 - CODAGE
 - TESTS UNITAIRES
 - INTEGRATION
 - TESTS ALPHA
 - TESTS BETA
 - RECETTE
 - MAINTENANCE
 - MIGRATION

Construire du logiciel

- **La gest. de projet**
 - Un projet : définition ?
 - Les grands éléments de la gestion de projet
 - Les méthodes de gestion et de suivi de projets informatiques
 - Les taux de « plantage » des projets informatiques
 - Les spécificités des projets informatiques

Construire du logiciel

• La gest. de projet

- Un projet : définition ?

Construire du logiciel

• La gest. de projet

- Les grands éléments de la gestion de projet

Construire du logiciel

• La gest. de projet

- Les méthodes de gestion et de suivi de projets informatiques
 - PERT/GANTT/Cycle en V
 - eXtrem Programing
 - Amazon.fr (mobipocket) Architecte et « Warrior »

Construire du logiciel

• La gest. de projet

- Les taux de « plantage » des projets informatiques
- Les spécificités des projets informatiques

Construire du logiciel

• La gest. de projet

- Selon The Standish Group International (2002)
 - 16 % des projets informatiques aboutissent,
 - 53 % sont remis en question,
 - 31 % échouent.

Source : <http://www.01net.com/article/212560.html>

Construire du logiciel

• La gest. de projet

- Selon Cap Gemini (2002),
 - 31 % des projets sont abandonnés avant terme,
 - 16 % ne remplissent pas les fonctions prévues
 - 53 % dépassent le budget initial.

Source :
<http://www.01net.com/article/212560.html>

Construire du logiciel

- **La gest. de projet**

Dans un projet, c'est l'architecture technique, véritable fondation du système, qui provoque le plus de dérapages »

Conduite de projet, une charte s'impose
<http://www.01net.com/article/212560.html>

Construire du logiciel

- **La gest. de projet**

- Les spécificités des projets informatiques

Construire du logiciel

- **L'estimation du projet**

- Comment estimer un projet ?

Construire du logiciel

- **Les tests**

- Quels types des tests ?
- Pour quoi faire ?
- Quand tester ?
- Qui est responsable des tests ?
- Comment tester l'intestable ?
- Le test comme philosophie (XP)

Construire du logiciel

- **Les tests**

- Black Box
- White Box
- Conformité
- Non-conformité
- Alpha
- Beta
- Unitaires
- Intégration
- Fonctionnels
- Intégration système
- Recette
- Non régression
- Double aveugle

Construire du logiciel

- **Le génie logiciel**

- Logiciel : modéliser – par un langage formel – une réalité (complexe)
- Génie logiciel = produire des logiciels de « qualité »

Construire du logiciel

• Le génie logiciel

- Validité (do the job)
- Robustesse
- Extensibilité
- Réutilisabilité
- Interopérabilité (API ouvertes)
- Optimisé (selon les ressources)
- Portabilité
- Vérifiabilité (logiciel facilement testable)
- Intégrité : (sécurité du logiciel par lui même)
- Ergonomie

Construire du logiciel

• PAQ & Plan de Configuration

- Définition ?
- Utile ou fumisterie ?

Log. O.O. = modélisation sociale

- Construire du logiciel = modéliser une organisation communicante
 - Logiciel à très haut niveau de complexité
 - Fini le logiciel linéaire et algorithme
 - Utilisation de framework de haut niveau
 - Faire du logiciel : c'est savoir choisir, assembler et faire communiquer des briques
 - Prise en compte d'exceptions, cas particulier...
 - Création de règles « humaines »
 - Ex : modélisez les règles de réduction pour un client fidèle dans une grande surface

Log. O.O. = modélisation sociale

- Construire du logiciel = modéliser une organisation communicante
 - Modélisons une (grosse et complexe) organisation communicante
 - Modélisons la création d'une grosse entreprise
 - Multinationale
- Vos propositions ?

Log. O.O. = modélisation sociale

• Notions

- Interface
 - Définir le savoir faire
 - Définir savoir communiquer
- Élément atomique
 - Encapsulation de la compétence
 - Cacher la complexité
 - Protéger ses attributs
- Héritage
 - Polymorphisme
- Vue multi-niveau
- Sémantique forte
 - Collaboration inter-service
- Notion de couplage (fort)
- Composition (contre héritage)
- Package
- Visibilité

Log. O.O. = modélisation sociale

• Notions

- Objets Distribués
 - Employé en « self service »
- Réaction en chaîne des messages
- MashUP

Le paradigme O.O

- Programmation par Interface (contrat)
- Visibilité (Public, Private, Protected, Package)
- Héritage & Polymorphisme
- Composition
- Encapsulation des attributs & Méthodes (set, get)
- Méthodes et classes Abstraites
- Les Packages
- Static

Le paradigme O.O

•Programmation par Interface (contrat)

-

Le paradigme O.O

•Visibilité

- Public,
- Private,
- Protected,
- Package

Le paradigme O.O

•Encapsulation des attributs

- Méthode *set* et *get*

Le paradigme O.O

•Héritage & Polymorphisme

- Héritage : le piège de l'empilement de concepts
- Héritage et spécialisation
- Le polymorphisme
 - Pilotage « universel »
 - Évolutivité simplifiée
- Factorisation
- Réutilisation (set, get, auto gestion erreur, auto définition...)

Le paradigme O.O

•Couplage

- Couplage faible
- Couplage fort

Le paradigme O.O

- **Agregation et Composition**

-

Le paradigme O.O

- **Méthodes et classes Abstraites**

.

Le paradigme O.O

- **Les Packages**

.

Le paradigme O.O

- **Propriétés *static***

-

Le paradigme O.O

- Collection et parcours
- Objets Bidon
- Garbage Collector
- Affectation par référence & Clonage (Profond)
- Persistance
- Fonction equal
- Introspection

Le paradigme O.O

- Binding (Mapping) OO/SGBD
- Introspection
- Pattern MVC
- Applications distribuées

Le paradigme O.O

- **Applications distribuées**

- JEE et EJB
- MashUP et Webservice