



Référence : STC - DOTNET - 70 - 549

5 jours - 35h

Microsoft 70-549

NIVEAU UTILISATEUR

Découverte

Initiation

Maîtrise

Expertise

Informations Coursus

Durée : 35 heures - 5 jours

Certification préparée : Microsoft 70-549

Durée de la certification : entre 2 et 4 heures

Public : Développeur, Webmaster, Architecte

Pré-requis

- Connaître les bases du .NET Framework
- Savoir utiliser Visual Studio 2005
- Expérience dans le développement d'applications d'entreprise
- Avoir suivi le cours 70-529 et 70-536

Contenu de la formation

Concevoir une application

- Evaluer la faisabilité du concept de l'application
- Recommander les technologies qui satisfont au mieux les besoins du projet
- Rechercher des solutions existantes à des problèmes similaires
- Evaluer les spécifications techniques de l'application pour vérifier que tous les besoins sont satisfaits
- Traduire ces spécifications en diagrammes ou pseudo-code

Concevoir la base de données

- Proposer un schéma de base de données
- Identifier les procédures stockées à développer sur la base de données

Etudier la conception logique et physique de l'application

- Du point de vue des performances
- Du point de vue de la facilité de maintenance
- Du point de vue de l'extensibilité
- Du point de vue de l'évolutivité.
- Du point de vue de la disponibilité.
- Du point de vue de la sécurité.
- Du point de vue de la récupération.
- Du point de vue de l'intégrité des données

Concevoir des composants

- Déterminer le nombre et le type de composants à développer
- Déterminer à quel niveau de l'application le composant doit se trouver
- Etablir le cycle de vie des composants
- Etudier le besoin d'utilisation d'un design pattern
- Décider de la création d'un prototype
- Ecrire la documentation du composant
- Architecturer l'API publique du composant

Développer les composants

- Déterminer quelles fonctionnalités doivent être héritées ou implémentées
- Déterminer les ressources requises (ressources non managées, threads, etc.)

Conception et développement d'une plateforme applicative

- Identifier les composants logiciels existants réutilisables
- Déterminer si ces composants doivent être étendus
- Décider si l'une des fonctionnalités existantes doit être cachée
- Tester le composant retenu contre les spécifications

Choisir une implémentation adéquate pour le modèle logique

- Sélectionner une méthode de stockage des données
- Sélectionner une structure appropriée pour le flux des données
- Sélectionner une technique de gestion des états
- Sélectionner une implémentation de la sécurité appropriée

Choisir une méthode de journalisation des événements

- Décider quelles seront les données à enregistrer
- Choisir une méthode de stockage pour le journal
- Déterminer si la journalisation doit être centralisée, distribuée, etc.

Audit de l'application

- Déterminer les besoins de l'application en termes d'audit et surveillance.
- Déterminer quels aspects de l'application doivent être audités ou surveillés : utilisation de la mémoire, sécurité, bugs, etc.
- Choisir un mécanisme d'enregistrement.
- Déterminer les niveaux de surveillance requis

Test et stabilisation d'une application

- Créer une stratégie de test
- Créer une stratégie de tests unitaires
- Créer une stratégie de tests d'intégration
- Créer une stratégie de tests de résistance
- Créer une stratégie de tests de performance
- Evaluer les spécifications de l'environnement de test
- Concevoir les tests unitaires
- Décrire les scénarii à tester
- Etablir la couverture nécessaire
- Mener les tests d'intégration
- Vérifier les résultats

Résoudre les bugs

- Rechercher les bugs identifiés
- Reproduire le bug
- Evaluer le cout associé à la correction du bug
- Corriger le bug

Déployer et supporter l'application

- Identifier les piques de performance
- Analyser des tendances
- Analyser les données recueillies par la surveillance et l'audit de l'application

Evaluer le plan de déploiement

- Identifier les dépendances de chaque composant au déploiement
- Identifier les scripts nécessaires au déploiement
- Identifier les scripts nécessaires à la base de données

Créer un diagramme de flux logique

- Evaluer la complexité des composants
- Evaluer la complexité des interactions avec les autres composants

Valider la configuration de l'environnement de production

- Vérifier les paramètres réseau
- Vérifier l'environnement de déploiement

Compétences acquises/ Objectifs

- Structuration d'une application en composants indépendants
- Evaluation des besoins de chacun des composants pour répondre aux spécifications de l'application
- Etablir une stratégie de développement, de test et de déploiement

Travaux Pratiques

- Etude de cas pratiques
- Rédaction de spécifications