



2025-11-14

Mise à jour automatique d'applications - questionnements et réflexions



Simon Hanot, Ayoub Khloifi, Boualem El
Guendouz
CEGEP DE CHICOUTIMI

Table des matières

Tâche 1 : Planifier les besoins de mise à jour de l'application	2
Question 1 : Est-ce qu'on désire que l'application informe l'utilisateur qu'une nouvelle version est disponible et que ce dernier aille la télécharger et l'installer lui-même?.....	2
Question 2 : Est-ce qu'un utilisateur peut utiliser l'application même si elle n'est plus à jour?.....	2
Question 3 : Est-ce qu'on désire offrir de la rétrocompatibilité entre les versions ?	2
Question 4 : Est-ce qu'on désire communiquer clairement les changements et nouveautés de chaque mise à jour à l'utilisateur ?.....	2
Question 5 : Est-ce que la mise à jour automatique doit se faire en arrière-plan sans interrompre l'utilisateur ?.....	3
Tâche 2 : Documenter les technologies disponibles	4
1. AutoUpdater.NET	4
2. wyBuild (wyUpdate / AutomaticUpdater)	4
3. Squirrel.Windows.....	5

Tâche 1 : Planifier les besoins de mise à jour de l'application

Question 1 : Est-ce qu'on désire que l'application informe l'utilisateur qu'une nouvelle version est disponible et que ce dernier aille la télécharger et l'installer lui-même?

Informer l'utilisateur qu'une nouvelle version est disponible et le laisser la télécharger puis l'installer lui-même est une approche courante. Cette méthode donne à l'utilisateur le contrôle du processus et permet d'éviter les installations inattendues.

Question 2 : Est-ce qu'un utilisateur peut utiliser l'application même si elle n'est plus à jour?

Oui, on peut utiliser une application même si elle n'est pas à jour, mais c'est risqué car elle peut être moins sûre, moins stable et certaines fonctions peuvent ne plus marcher. C'est toujours mieux de la mettre à jour pour éviter ces problèmes.

Question 3 : Est-ce qu'on désire offrir de la rétrocompatibilité entre les versions ?

Oui, la rétrocompatibilité permet aux nouvelles versions d'un logiciel de fonctionner avec les anciennes, assurant ainsi une transition sans perte de données pour l'utilisateur. C'est une garantie de confort et de fidélisation. Elle est donc généralement souhaitée lors de l'évolution des versions.

Question 4 : Est-ce qu'on désire communiquer clairement les changements et nouveautés de chaque mise à jour à l'utilisateur ?

Il est important de communiquer clairement les changements et nouveautés lors de chaque mise à jour pour maintenir la transparence avec les utilisateurs.

Cependant, dans certains cas, on peut décider de ne pas trop insister sur ces informations si l'on pense qu'elles risquent de dérouter ou de submerger l'utilisateur avec des détails jugés non essentiels.

Question 5 : Est-ce que la mise à jour automatique doit se faire en arrière-plan sans interrompre l'utilisateur ?

La mise à jour automatique en arrière-plan doit idéalement se faire sans interrompre l'utilisateur afin d'assurer une expérience fluide et agréable. Cela permet à l'application d'être constamment à jour, garantissant ainsi la sécurité et la stabilité sans perturber les activités en cours. Néanmoins, il convient de veiller à ce que cette opération ne consomme pas excessivement les ressources telles que la batterie ou la connexion Internet. Par ailleurs, il est conseillé d'offrir à l'utilisateur la possibilité de gérer cette fonctionnalité selon ses préférences, afin de préserver son contrôle et d'éviter toute frustration. Cette approche équilibre sécurité, performance et respect de l'expérience utilisateur.

Tâche 2 : Documenter les technologies disponibles

1. AutoUpdater.NET

URL: <https://github.com/ravibpatel/AutoUpdater.NET>

Pourquoi je la trouve intéressante :

- Bibliothèque open-source d'applications .NET permettant d'ajouter facilement une logique de mise à jour automatique dans une application desktop
- Elle supporte les versions récentes de .NET

Types de projet pour lesquels elle est destinée :

Applications desktop Windows (WinForms, WPF) codées en C# (ou VB.NET).

2. wyBuild (wyUpdate / AutomaticUpdater)

URL: <Automatic updating & make patches for your C#, VB.NET, C, C++, apps - wyBuild>

Pourquoi je la trouve intéressante :

- Solution “clé en main” pour créer des mises à jour automatiques. On inclut un “updater” dans l'application, on déclenche la mise à jour automatique ou semi-automatique.
- Génère des « tiny update patches » qui réduisent la taille des mises à jour, optimisant ainsi la bande passante utilisée et accélérant le processus de déploiement.
- Adaptée pour les applications écrites en .NET.

Types de projet pour lesquels elle est destinée :

Idéal pour une application native Windows (C#, VB.NET, etc.) qui veut proposer une mise à jour automatique intégrée.

3. Squirrel.Windows

URL: <https://github.com/Squirrel/Squirrel.Windows>

Pourquoi je la trouve intéressante :

- Solution moderne pour les mises à jour automatiques silencieuses des applications Windows
- Optimise l'utilisation de la bande passante grâce aux mises à jour différencielles
- Déploiement rapide, silencieux et transparent pour une meilleure expérience utilisateur
- Utilisée par des produits reconnus comme Microsoft Teams, assurant fiabilité et robustesse

Types de projet pour lesquels elle est destinée :

Applications bureau Windows développées en C# ou autres langages .NET