**Scénarios et tactiques de Modifiabilité**

[1.Interface utilisateur](#_30j0zll)

[Tactique: Augmenter la cohérence sémantique](#_1fob9te)

[2.Ajout d’un nouveau type de transaction (Quelle transaction exactement?)](#_3znysh7)

[Tactique: Reduce coupling (Encapsulation)](#_2et92p0)

[3.Démarrage du guichet](#_tyjcwt)

[Tactique: Binding time of modification](#_3dy6vkm)

[4.Distribution des nouveaux billets de banqueAvoir un distributeur d’argent modulaire.](#_1t3h5sf)

[Tactique: Refactor](#_4d34og8)

[5.Dépôt d’argent de devise étrangère](#_2s8eyo1)

[Tactique: Refactor](#_35nkun2)

[6.Conception de nouveaux guichets automatiques](#_1ksv4uv)

[Tactique: Split Modules](#_44sinio)

[Tactique: Refactor](#_2jxsxqh)

[7.Migration des OS des guichets automatiques](#_z337ya)

[Tactique: Split Modules](#_3j2qqm3)

[Tactique: Reduce coupling](#_1y810tw)

[8.Rendre le GAB compatible avec la technologie sans-fil utilisée dans les modes de paiement PayPass.](#_4i7ojhp)

[Tactique: Couplage](#_2xcytpi)

[Tactique: Séparation des modules](#_1ci93xb)

[9.Un nouveau type de carte est ajouté.](#_3whwml4)

[Tactique: Abstract Common Services](#_2bn6wsx)

| **Scénario** | 1.Interface utilisateur |
| --- | --- |
| **objectifs d'affaires** | Offrir une interface conviviale et simple d’utilisation sans affecter la performance. |
| **Source** | Développeur |
| **Stimulus** | modification des fonctionnalités des boutons |
| **Artéfact** | Interface |
| **Environnement** | Design time |
| **Réponse** | * Faire la modification * Tester la modification * Déployer la modification |
| **Mesure de la      réponse** | * Effort fourni dans la modification * Impact de la modification sur la performance du GAB * Temps nécessaire pour les modifications futurs |
| **Questions** | 1. Est-ce que cette modification est nécessaire ? quelle est sa valeur ajoutée ? 2. Quelles sont les dépendances entre les modules affectés par cette modification ? 3. Quel est le degré de modifiabilité de l’interface ? 4. quel est le coût de cette modification ? 5. es que l’interface risque de changer souvent ? 6. faut-il modifier directement ou implémenter un mécanisme de modifiabilité ? |

### Tactique: Augmenter la cohérence sémantique

**Description**: Augmenter la cohérence sémantique nous permettra de distinguer plus facilement les fonctionnalités des modules.

**Justification**:

justification de l’implémentation d’un mécanisme de modifiabilité:

N\*Coût sans mécanisme  <  Coût mécanisme + N\*Coût avec mécanisme

choix de la tactique:

Cette tactique nous permettra de répondre plus facilement à des questions comme :quels fonctions sont exécutées dans quel boutons ?

le fait de voir le design du bouton et ce qui est exécutée dans ce boutons et/ tout les événement/listener qui lui sont lié d’un manière distingué , facilite le processus de modifiabilité des interfaces.

| **Scénario** | 2.Ajout d’un nouveau type de transaction |
| --- | --- |
| **objectifs d'affaires** | Pouvoir ajouter un nouveau type de transaction au logiciel en moin de 20 jours de développement pour 2 développeurs. |
| **Source** | Le développeur |
| **Stimulus** | Ajouter un nouveau type de transaction |
| **Artéfact** | Code source |
| **Environnement** | Build time |
| **Réponse** | - Programmer la nouvelle transaction  - Tester la nouvelle transaction  - Déployer la nouvelle transaction |
| **Mesure de la réponse** | - Nombre d’heure par développeur  - Nombre de dépendances sur le nouveau code |
| **Questions** | 1. Est-ce que ce changement affecte le fonctionnement des autres types de transactions? 2. Est-ce que le changement requiert un changement dans le processus global d’une transaction? 3. Est-ce que la performance sera affectée par cet ajout? |

### Tactique: Reduce coupling (Encapsulation)

**Description**: Introduction d’une interface explicite pour l’ajout de différents modules. Permets de réduire le risque qu’un changement affecte d’autres modules. Cette interface doit être abstraite.

**Justification**: Une interface de type Transaction faciliterait le processus pour effectuer différents types de processus. Toutes les transactions auraient le même processus et ils n'affectent donc pas le processus des autres. Les responsabilités externes des modules peuvent seulement être utilisées à travers l’interface.

| **Scénario** | 3.Démarrage du guichet |
| --- | --- |
| **objectifs d'affaires** | Pouvoir ajouter des modules au démarrage et qu’elles soient transparente pour le reste du système du guichet. |
| **Source** | Le développeur |
| **Stimulus** | Ajouter des modules lors du démarrage du guichet |
| **Artéfact** | Code source |
| **Environnement** | Build time |
| **Réponse** | - Programmer le nouveau module  - Tester le nouveau module  - Déployer le nouveau système sur les guichets. |
| **Mesure de la réponse** | - Nombre d’heure par employé  - Complexité de la programmation de cet ajout |
| **Questions** | 1. Est-ce que l'ajout de module va affecter la stabilité du guichet actuel? 2. Est-ce que les performances vont être affectées significativement? |

### Tactique: Binding time of modification

**Description**: Une architecture qui est bien développée accommodera les modifications tard dans le cycle de développement du projet. En moyenne, le coût de modification tard dans le cycle de développement sera moindre et même très près de zéro.

**Justification**: Dans ce cas, la maintenabilité du système de démarrage sera moins coûteuse et plus facile à modifier. Il sera très rapide à ajouter des modules lors du démarrage si l’architecture a été pensée pour la modificabilité avec cette tactique.

| **Scénario** | 4.Avoir un distributeur d’argent modulaire. |
| --- | --- |
| **objectifs d'affaires** | Simplifier la compatibilité avec les nouveaux billets |
| **Source** | * développeur d’applications, * administrateur système |
| **Stimulus** | * Déploiement de nouveaux billets faits d’un matériau incompatible avec les distributeur actuel. |
| **Artéfact** | * distributeur d’argent |
| **Environnement** | * Opération normale |
| **Réponse** | * Affichage d’un message de confirmation * ecriture de log durant un retrait |
| **Mesure de la réponse** | * Alimenter le distributeur (argent) avec tous les types d'argent |
| **Questions** | 1. Quel changement physique sera fait sur le distributeur actuel ? 2. Quelle option serait modifiée au niveau de l’application existante afin de répondre à cette fonctionnalité ? |

### Tactique: Refactor

**Description:** Il faut ajouter le changement sans impacter le fonctionnement externe qui existait au préalable.

**Justification:** Il faut que le changement soit modulaire, incremental.

| **Scénario** | 5.Dépôt d’argent de devise étrangère |
| --- | --- |
| **objectifs d'affaires** | Permettre la conversion de devise étrangère |
| **Source** |  |
| **Stimulus** | * Mise en place d’une fonctionnalité permettant au client de faire un dépôt de devise étrangère sur son compte |
| **Artéfact** | * distributeur d’argent |
| **Environnement** | * Opération normale |
| **Réponse** | * Le système affiche la conversion * Le système crédite le compte du client du montant déposé * Affichage d’un message de confirmation * Impression d’un reçu |
| **Mesure de la réponse** | * Le temps que met le système pour créditer le compte du client * Combien d’options faut-il d’abord choisir avant de déposer une devise étrangère ? |
| **Questions** | 1. Comment est gérée la conversion entre la devise étrangère et la monnaie locale ? 2. Comment le système peut-il vérifier l’authenticité de tous les billets déposés (vrai ou contrefaçon) ? 3. Quelle type de modification (matérielle ou logicielle) faut-il faire sur le système existant afin d’ajouter ce module de conversion ? 4. Combien coûtera la mise en place de ce module ? |

### Tactique: Refactor

**Description:** Il faut ajouter le changement sans impacter le fonctionnement externe qui existait au préalable.

**Justification:** Il faut que le changement soit modulaire, incremental.

| **Scénario** | 6.Conception de nouveaux guichets automatiques |
| --- | --- |
| **objectifs d'affaires** | Intégration de nouveaux guichets automatiques disposant d’un nouvel OS.Mettre à jour les application du serveur central pour qu’il puisse prendre en charge les nouveaux guichets et permettre la communication entre les nouveaux et les anciens guichets |
| **Source** | Développeurs de l’application |
| **Stimulus** | Modification de la technologie utilisée sur les guichets automatiques. |
| **Artéfact** | Code source et configurations du serveur central |
| **Environnement** | Temps de conception |
| **Réponse** | Modification de déploiement  Modification des tests |
| **Mesure de la réponse** | Coût de déploiement  Temps requis pour la conception  Nombre de nouveaux défauts introduit dans le système  Nombre d’artéfacts affectées par l’intégration des nouveaux guichets |
| **Questions** | 1. Qui est responsable pour l’installation des guichets? 2. Comment est-ce qu’on testera les guichets? 3. Comment les nouvelles applications du serveur central seront-elles testées? |

### Tactique: Split Modules

**Description**: Faire des parties des modules existants

**Justification**: Isoler les parties dans les modules qui doivent être modifiées. Ceci va permettre d’effectuer les modifications nécessaires pour intégrations des modules avec les nouveaux guichets sans affecter les autres parties des modules à modifier.

### Tactique: Refactor

**Description**: Enlever code dédoublé ou inutile

**Justification**: Suite à la modification des modules, il est utile de réviser le code et d’enlever l’utilisation de code inutile ou de réduire le nombre de lignes de code en faisant la création de méthode si nécessaire. Le refactoring du code facilite la modification des modules dans le futur et améliore la modifiabilité.

| **Scénario** | 7.Migration des OS des guichets automatiques |
| --- | --- |
| **objectifs d'affaires** | Mettre à jour les OS des anciens guichets automatiques pour augmenter la sécurité de l’application |
| **Source** | Développeurs de l’application |
| **Stimulus** | Modification du système utilisé sur les machines (guichets) |
| **Artéfact** | Code source, interface |
| **Environnement** | Temps de déploiement |
| **Réponse** | Modification de déploiement  Modification des tests |
| **Mesure de la réponse** | Coût de déploiement  Nombre de nouveaux défauts introduit dans le système  Nombre de défauts corrigés dans le système  Nombre d’artéfacts affectées par l’intégration du nouvel OS sur les guichets |
| **Questions** | 1. Qui est responsable de la migration des OS? 2. Quel est le temps requis pour procéder à la migration? 3. Comment tester la sécurité de l’application? 4. Peut-on prévoir un “roll-back” en cas d’échec de la migration? |

### Tactique: Split Modules

**Description**: Faire des parties des modules existants

**Justification**: Isoler les parties dans les modules qui doivent être modifiées. Ceci va permettre d’effectuer les modifications nécessaires pour intégrations des modules avec les nouveaux guichets sans affecter les autres parties des modules à modifier.

### Tactique: Reduce coupling

**Description**: Réduire le couplage entre les paquets et les composants de l’application

**Justification**: Cette mise à jour va permettre de réduire les couplages contenant dans l’architecture de l’application des guichets.

| **Scénario** | 8.Rendre le GAB compatible avec la technologie sans-fil utilisée dans les modes de paiement PayPass. |
| --- | --- |
| **Objectifs d'affaires** | Concevoir une architecture qui facilite le support incrémental des technologies sans-fil (maximum de 3 mois après leur commercialisation) |
| **Source** | Le directeur |
| **Stimulus** | La direction veut ajouter la capacité au GAB de prendre en charge les technologies PayPass pour authentifier les cartes des usagers. |
| **Artéfact** | Interfaces des composantes physiques et protocol de communication entre carte et guichet. |
| **Environnement** | Runtime |
| **Réponse** | Effectuer les modifications:   * Utiliser une architecture en couche pour faciliter la modification de notre système et supporter les technologies futures * En utilisant le patron Stratégie, on peut définir un contrat pour faire l’implémentation de toutes les différentes technologies mobiles en utilisant les interfaces.   Tester les modifications:   * Faire des tests.   + Des tests pour simuler l’implémentation des différentes technologies   + Des tests unitaires   + Des tests d’intégration   Déploiement des modifications:   * Durant la nuit, entre 3h00 et 4h00 A.M., via processus de mises à jour en utilisant le réseau existant. |
| **Mesure de la réponse** | 1 000 000 CAD$. Ces dépenses sont prévues pour le travail d’une équipe de 10 personnes mobilisées pendant un an avec un salaire moyen de 70 000$/an par personne avec une contingence de +/- 30%. |
| **Questions** | 1. Quand le changement peut-il être fait ? 2. Qui peut faire le changement ? 3. Avec le coût de modification, est-ce que c’est rentable ? 4. Comment effectuer la mise-à-jour du GAB ? 5. Quand faire la mise-à-jour du GAB ? |

### Tactique: Couplage

**Description**: La réduction du couplage entre deux modules diminue les coûts de modifications potentiels pour ces deux modules.

**Justification**: Comme nous souhaitons ajouter régulièrement et rapidement de nouvelles technologies, il est important qu’il y ait un minimum de couplage avec ces modules.

### Tactique: Séparation des modules

**Description**: Des modules plus petits sont généralement associés à des coûts de modifications plus bas.

**Justification**: Comme nous souhaitons ajouter régulièrement et rapidement de nouvelles technologies, il est important d’avoir un minimum de code pour chacune des technologies via l’utilisation de classes abstraites ou d’interfaces.

| **Scénario** | 9.Un nouveau type de carte est ajouté. |
| --- | --- |
| **objectifs d'affaires** | Supporter nouveaux types de cartes en moins de 2 mois |
| **Source** | La compagnie de la nouvelle carte |
| **Stimulus** | Une demande est faite pour supporter le nouveau type de carte |
| **Artéfact** | Lecteur de carte, code |
| **Environnement** | Design time |
| **Réponse** | Un membre de la compagnie de la carte explique les spécificité de la carte au responsable de la modification  Une équipe de développeur est assigné à la modification  Les lecteur de carte sont modifié au besoins |
| **Mesure de la réponse** | Délais avant l’implémentation de la modification. |
| **Questions** | 1. Quel est le temps moyen requis pour l’implémentation d’un nouveau type de carte? 2. Quelles sont les classes affectées par un changement de technologie de cartes? 3. Combien de cartes sont supportées par le système GAB? |

### Tactique: Abstract Common Services

**Description:** Les services de lecture des types de carte sont rendu abstrait.

**Justification:** En ayant une base commune abstraite, il est plus facile d'ajouter et supprimer un type carte.