

Rep++, un tremplin vers une plateforme *low-code*

Les applications telles que nous les connaissons aujourd'hui, qui prennent des mois voire des années à développer, à partir de spécifications aussi nombreuses que rigides, et qui ne ciblent que les postes de travail traditionnels sont en net déclin. Elles sont remplacées par des applications plus légères, plus flexibles, mieux adaptées, qui peuvent être déployées plus rapidement¹. Cette nouvelle façon de faire s'implante résolument, s'impose comme un standard et pousse vers le haut la cadence de livraison et de déploiement.

Pour livrer plus rapidement et gagner ou conserver un avantage concurrentiel, les entreprises seront obligées de faire évoluer leur processus de développement d'applications vers une approche plus souple et plus productive. Pour y parvenir, les entreprises devront:

- Accélérer le rythme de production et de livraison de leurs applications.
- Créer des applications fonctionnant sous diverses architectures pour des dispositifs variés.

Les plateformes *low-code*

Les plateformes *low-code* gagnent en popularité en raison de leur flexibilité et de leur capacité à livrer rapidement, caractéristiques au cœur de ce nouveau paradigme de développement d'applications. Les plateformes *low-code* sont des environnements de développement qui « ... abaissent les barrières entre les spécifications et le déploiement »² en :

- minimisant la programmation manuelle;
- proposant une plateforme unique permettant de partager les spécifications (métadonnées), la conception, le développement et le déploiement;
- fournissant un environnement centralisé qui accroît l'automatisation de la configuration, de la maintenance et du déploiement.

Le développement *low-code* non seulement partage les caractéristiques principales de l'approche agile, telles que les cycles de développement itératifs rapides et les livraisons fréquentes, mais va au-delà de cette approche³. Le développement *low-code* préconise:

- Une réduction substantielle de la programmation manuelle. Cette réduction est réalisée par l'utilisation de « modèles de données facilement configurables qui éliminent les difficultés liées à l'intégration des données », et d'outils pour définir et assembler rapidement les applications⁴.
- Une capacité d'expérimentation accrue. L'approche itérative facilite l'exploration d'idées nouvelles et de solutions créatives, et permet d'apprendre de leurs résultats⁵. L'expérimentation peut non seulement conduire à des innovations inattendues, mais également réduire la résistance au changement qui est si caractéristique des grands projets rigides et surplanifiés.
- Des tests continus et une rétroaction intégrée au processus. En particulier, l'implication étroite du client, idéalement dès le début du projet, et une rétroaction rapide de ce dernier, fait fondamentalement partie du processus⁶. Les itérations rapides démontrent les progrès, permettent de déterminer les améliorations possibles ou de repérer les points problématiques, et aident tous les participants à valider et à clarifier les spécifications en produisant des prototypes fonctionnels de façon incrémentale. Puisque le code est livré avec de légères améliorations à chaque itération, il peut être corrigé rapidement et plus facilement, ce qui réduit les risques tout en assurant la qualité du code produit.

³ Richardson et Rymer, p. 12.

⁴ Richardson et Rymer, p. 3.

⁵ Forrester Consulting, p. 10.

⁶ Forrester Consulting, p. 6-7.

¹ Forrester Consulting, p. 2.

² Richardson et Rymer, p. 3-4.

- Une capacité de livraison et déploiement continus grâce à l'automatisation. Le déploiement peut être une étape terrifiante, notamment avec les applications multiplateformes: les garder toutes synchronisées nécessite une coordination exceptionnelle qui est difficile à atteindre. L'étape de déploiement, lorsqu'incluse dans le cycle, devient partie intégrante du processus et peut ainsi être automatisée et testée comme le reste de l'application⁷; finalement, elle est mieux comprise et mieux maîtrisée⁸. Les applications sont livrées plus rapidement et de manière plus fiable, et ne nécessitent qu'une intervention manuelle minimale⁹.
- Des applications adaptées aux diverses plateformes mobiles. Les clients l'exigent, les entreprises qui les livrent en bénéficient.¹⁰

Les plateformes low-code offrent de nombreux avantages.

- Elles soutiennent des cycles de prototypage plus fréquents qui produisent des livrables dont les incréments sont moins importants.
- Elles aident à mettre en place des pratiques plus productives qui comprennent des équipes interfonctionnelles pouvant gérer des projets complets, de la conception à la livraison¹¹.
- Elles inscrivent les efforts de collaboration comme un aspect inhérent au processus.
- Elles encouragent le déploiement accéléré.
- Elles centralisent le modèle, la gestion et la configuration pour faciliter le développement en équipe et la livraison des applications sur de multiples plateformes.
- Elles permettent d'implanter de meilleures pratiques de développement telles que (a) la séparation des préoccupations par le découplage des couches d'une application pour permettre des changements indépendants, et (b) l'augmentation de l'automatisation pour améliorer la cohérence et la qualité des applications.^{12,13}

En somme, les plateformes low-code nourrissent des pratiques de développement et de déploiement plus productives, qui peuvent démontrer rapidement la valeur d'un projet.

Comment Rep++ peut vous mener vers un développement low-code

Rep++ s'inscrit directement dans l'optique des plateformes low-code, qui permettent aux entreprises d'atteindre les objectifs de livraison et de déploiement continus.

Rep++ réduit intrinsèquement la codification manuelle grâce à son modèle Rep++ et à son mécanisme d'exécution de modèle.

- Le modèle Rep++ est conçu pour définir, grouper et organiser les métadonnées en un tout cohérent. Il est aisément configurable et compréhensible par tous les intervenants autorisés, et ne contient aucun code. Le modèle est centralisé et partagé avec tous ceux qui participent au projet, ce qui assure la cohésion des applications, du concept au déploiement.
- Le modèle Rep++ maximise la réutilisation. Par exemple, Rep++ gère toutes les occurrences d'un même champ dans les différents formulaires; sa définition est simplement réutilisée partout où elle est nécessaire. Une définition unique permet non seulement de maximiser la réutilisation, mais ultimement de réduire la maintenance.
- Le mécanisme d'exécution de modèle réduit nettement la programmation puisqu'au lieu d'utiliser du code, par exemple pour valider une donnée, une application utilise plutôt un méta-objet qui, à l'exécution, accède directement au modèle et aux informations dont elle a besoin. Ce mécanisme signifie également que tout changement apporté au modèle est automatiquement appliqué aux applications, sans codification additionnelle, et souvent sans recompilation ni autre intervention.

Rep++ automatise systématiquement le processus de développement, de la définition du modèle jusqu'au déploiement des applications.

- Les cadres applicatifs Rep++ minimisent la codification manuelle en automatisant une large portion des applications pour les diverses architectures et technologies. Qu'il s'agisse d'ASP/MVC, de WPF, de SPA ou autre, les fonctions complexes et répétitives telles que la navigation, la création, la modification, la gestion de transaction, la gestion des erreurs, et de nombreuses autres tâches sont prises en charge automatiquement. Les développeurs peuvent ainsi se concentrer sur la logique d'affaires de l'application.

⁷ Bittner et O'Donnell, p. 6.

⁸ Bittner et O'Donnell, p. 12.

⁹ Evans Research, p. 2.

¹⁰ Richardson et Rymer, p. 3.

¹¹ Bittner, p. 17.

¹² Bittner, p. 25.

¹³ Bittner et O'Donnell, p. 9.

- Les assistants Rep++ vous permettent de créer des applications visant des architectures variées — de simples applications clients-serveurs dont tous les objets se trouvent dans le même assemblage, aux applications multiniveaux ou services, dont les objets d'affaires résident sur différentes couches et les données transitent par l'entremise d'entités POCO entre les différents niveaux. Les assistants se chargent de définir les objets pour chacune des couches.
- Rep++ prend en charge et optimise automatiquement la couche d'accès aux données, pour les bases de données relationnelles les plus populaires, peu importe l'architecture.
- Rep++ crée automatiquement la couche de présentation et presque toutes les couches d'affaires et de services, selon la technologie et l'architecture désirées.
- Le déploiement continu consiste simplement à mettre à jour le référentiel contenant le modèle. Dans certains cas une recompilation est requise, sinon un redémarrage de l'application est suffisant.
- Un seul modèle est utilisé pour créer la même application pour diverses architectures et technologies, que ce soit pour un poste de travail, pour le web ou pour un dispositif mobile.

Rep++ facilite l'expérimentation de nouvelles idées et concepts. La modification du modèle ou l'intégration de nouvelles spécifications lors d'un projet ne deviennent plus des sources de tension, mais plutôt des sources d'innovation.

Rep++ favorise le contrôle continu et encourage les commentaires des clients, employés et partenaires. Comme le modèle peut être partagé facilement avec tous les participants, y compris le client, une rétroaction rapide est facile à réaliser, même à distance.

Rep++ permet de mettre en place de meilleures pratiques de développement, telles que :

- La séparation des préoccupations. Les objets de présentation et d'affaires sont habilement conçus pour leur couche respective.
- L'élimination de la réplication de code. Une définition unique pour représenter un élément

d'information réduit les ambiguïtés et facilite la maintenance. Rep++ applique ce principe de façon intrinsèque grâce à son modèle Rep++, où les métadonnées sont centralisées et réutilisées au lieu d'être répliquées. D'autres caractéristiques des objets Rep++, telles que les méthodes de code personnalisé et les événements Rep++, préviennent également la réplication du code.

En résumé

Pour livrer plus rapidement des applications de meilleure qualité et acquérir ou conserver un avantage concurrentiel, les entreprises doivent dès maintenant adopter une approche de développement plus légère et plus productive. La plateforme de développement low-code Rep++ permet de restructurer la stratégie de développement vers le déploiement continu. Parce qu'il présente une capacité d'automatisation inégalée, minimise la programmation manuelle et abaisse la résistance aux changements, Rep++ représente aujourd'hui l'une des meilleures plateformes low-code sur le marché.

Bibliographie

Bittner, K. *The Business of Continuous Delivery*, Forrester Research, Keynote address notes, CD Summit, New York, NY, June 19, 2014.

Bittner K. et G. O'Donnell. *The Eight Tenets of Faster Application Delivery*, Forrester Research, April 15, 2014.

Evans Research. *Continuous Delivery: The New Normal for Software Development*, Findings from Survey of Software Development Professionals Commissioned by Perforce Software, 2014.

Forrester Consulting. *The State of Modern Product Delivery; Challenges and Trends Making Product Delivery Responsible, Iterative, and Collaborative In The Age Of The Customer*, Thought Leadership Paper Commission by Jama Software, November, 2013.

Richardson, C. et J.R. Rymer. *New Development Platforms Emerge For Customer-Facing Applications*, Forrester Research, June 9, 2014.

