

Mettre le Low-code / No-code au service de vos projets innovants



1

Qu'est-ce que le Low-code / No-code ?

2

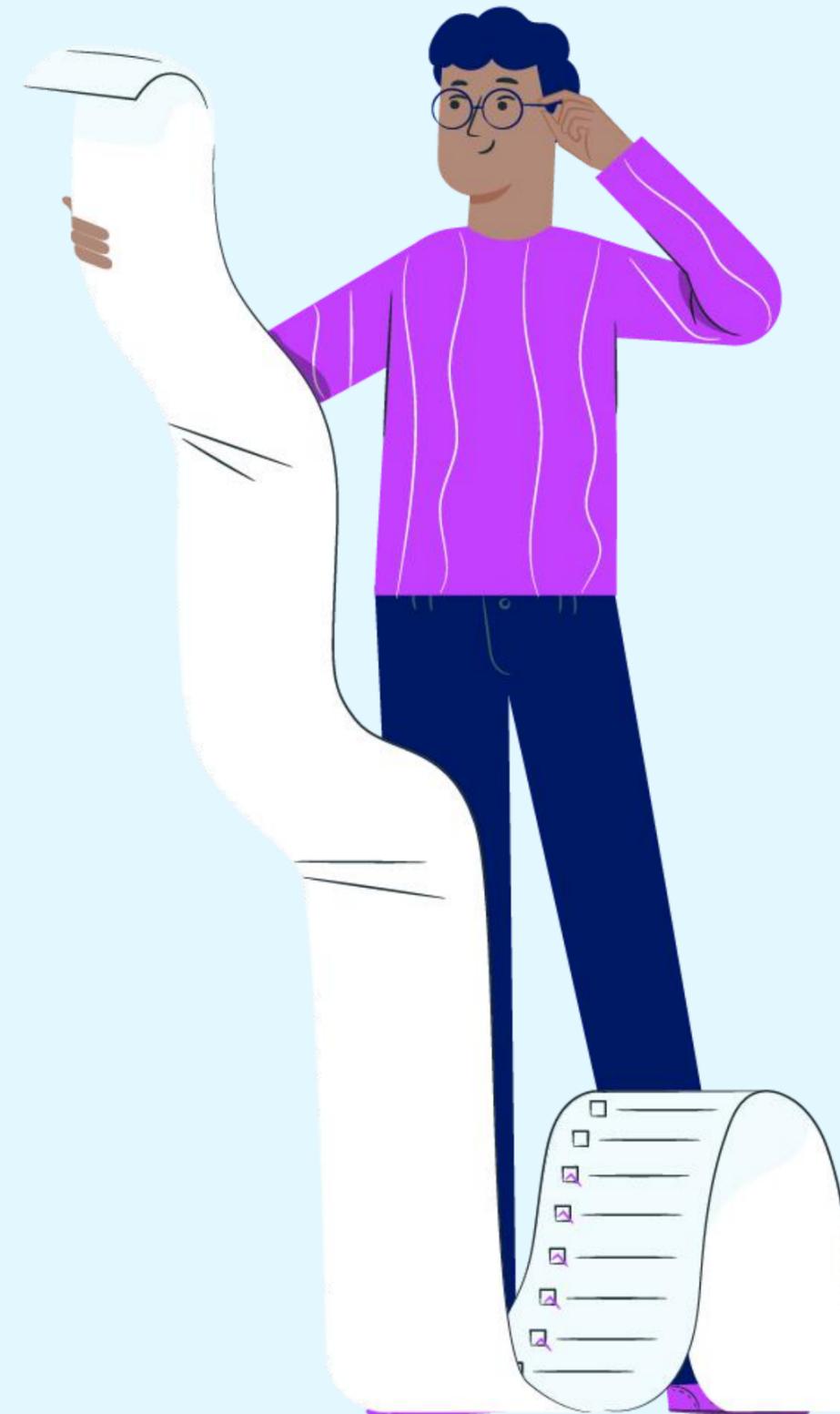
Le Low-code / No-code pour dérisquer vos projets

3

Quels outils, pour quelles problématiques ?

4

Mise en place dans votre organisation



Les outils de Low-code / No-code sont des plateformes permettant de créer des solutions digitales sans coder ou peu

Qu'est-ce que le Low-code / No-code ?

Un ensemble de plateformes qui offrent des “fonctionnalités de développement rapide” dédiées à la création, au déploiement et à l'exécution d'applications dans le cloud.

- Par définition, ces outils simplifient le développement d'applications. Les outils No-code permettent à tout un chacun de créer une solution digitale et de la confronter à son marché.
- Aujourd'hui, la majorité des logiciels No-code proposent des options “Low-code” en créant des outils permettant aux développeurs d'ajouter du code et ainsi rendre leur produit plus personnalisable.
- Cependant, les outils Low-code se démarquent par des fonctionnalités plus avancées et de plus grandes possibilités de personnalisation. Mais, ils ne sont pas utilisables sans le support des équipes de développement.

Le Low-code / No-code n'est pas...

- Un outil de **prototypage** (ex : Figma...)
- Une plateforme de **conception graphique** (Canva...)

Quelques exemples de plateformes Low-code / No-code

.bubble



zapier



WIX.com

Les principaux avantages du Low-code / No-code : accélération du time to market des solutions digitales et réduction des coûts de développement

Les outils Low-code / No-code répondent principalement à 2 grands besoins organisationnels...



Donner aux équipes métiers une alternative au "Shadow IT". Les collaborateurs « métiers » doivent pouvoir **tester leurs idées de solutions de manière simple**, via des outils accessibles et ergonomiques.



Concentrer les efforts des équipes IT sur les applicatifs à forte valeur ajoutée pour l'entreprise tout en fournissant une solution de développement rapide pour les autres besoins applicatifs.

...et présentent donc 4 avantages majeurs.



Accélérer le time to market des solutions digitales



Optimiser les coûts de conception et de développement



Pallier au manque de ressources de développement en interne



Basculer vers du SaaS ou du Cloud

L'adoption des outils Low-code / No-code continue de croître même si certains défis restent à relever

Les outils Low-code / No-code vont continuer à se démocratiser...

70%

des nouvelles applications développées par les entreprises utiliseront des technologies Low-code / No-code d'ici 2025, contre moins de 25 % en 2020.

Source : Gartner, 2021

13,8 milliards de dollars de revenus en 2021 pour le Low-code / No-code dans le monde, soit une augmentation de **22,6 %** par rapport à 2020.

Source : Havas, 2021

...afin de répondre aux différentes attentes clients...

90% : la réduction du temps de développement grâce aux solutions Low-code / No-code.

Source : Redhat, 2018

85% : des utilisateurs de plateformes Low-code / No-code affirment qu'elles ajoutent de la valeur à leurs projets.

0,5% : de la population mondiale sait aujourd'hui comment coder.

Source : Havas

...mais devront encore relever certains défis.

Manque de connaissance

47%

des entreprises ne sont pas au fait de l'existence de solutions Low-code / No-code.

Source : G2, 2021

Sécurité

25%

des entreprises pensent que les solutions Low-code / No-code sont ouvertes aux menaces et aux risques de sécurité.

Source : G2, 2021

1

Qu'est-ce que le Low-code / No-code ?

2

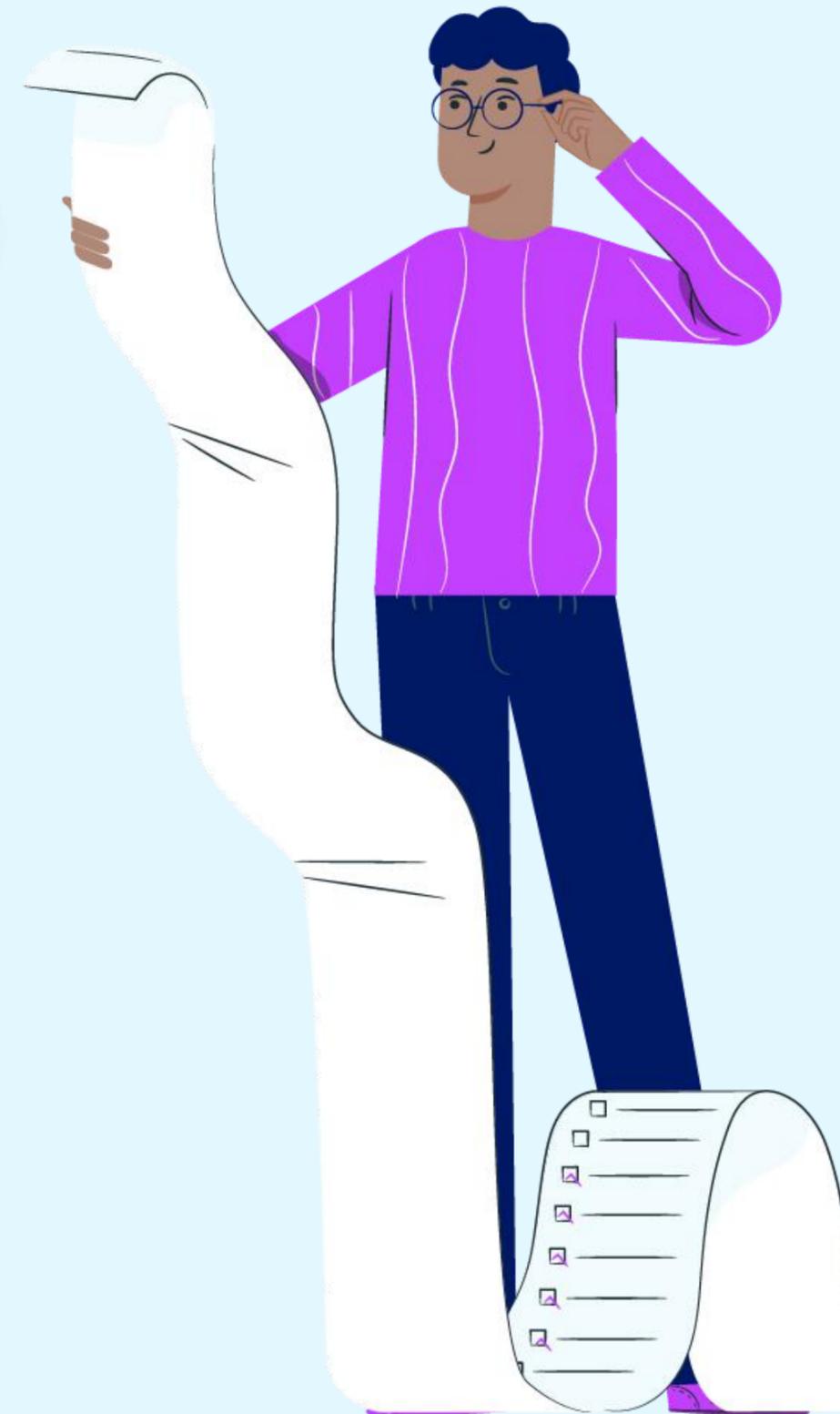
Le Low-code / No-code pour dérisquer vos projets

3

Quels outils, pour quelles problématiques ?

4

Mise en place dans votre organisation



Start-ups, grands groupes, ETI ou encore associations utilisent le Low-code / No-code pour leurs projets



50%* des moyennes et grandes entreprises françaises utilisent déjà le Low-code / No-code.

Deux cas réels : **SNCF** a fait appel à **Power Platform*** afin d'accroître l'efficacité opérationnelle de ses 270 000 salariés. L'objectif était ici de développer la culture digitale dans l'entreprise en utilisant la tech au services des employés. **Bolloré Transport & Logistics** a développé pour le Gabon une application de tracking en temps réel avec Power Platform afin d'optimiser la position des conteneurs et de gérer les stocks de manière optimale dans les différentes régions du pays.



Les entrepreneurs s'emparent du Low-code / No-code pour atteindre leur product market fit.

Un cas réel : La start-up anglaise **Treepoints** a créé un simulateur pour calculer ses émissions carbone puis éventuellement les compenser en finançant des projets à impact. Les fondateurs n'ayant pas d'expertise en codage et programmation se sont tournés vers la plateforme **Bubble****, qui leur a permis de mener à bien leur projet mais également de mettre à jour régulièrement la page des comptes utilisateurs.



Les associations et ONG utilisent le Low-code / No-code pour mener à bien leurs missions.

Un cas réel : L'association à but non lucratif **New incentives** collecte des dons afin de favoriser et accroître la vaccination infantile au Nigéria. **New incentives utilise AppSheet***** afin que l'accès à l'information soit simplifié mais également dans l'objectif de gérer l'alimentation des données. À savoir, localiser les besoins et organiser la logistique pour y répondre.

Microsoft Power Platform / Power Apps : solution de création d'applications à usage interne, intégrée à l'environnement Microsoft

** Bubble : solution intégrée pour la création d'applications mobile et web (SaaS, Marketplace...)

*** AppSheet : solution de création d'applications mobile et web intégrées à Gmail, Google Sheet et Google Space.

Le Low-code / No-code permet de matérialiser des concepts à moindre coût et les confronter rapidement à la réalité



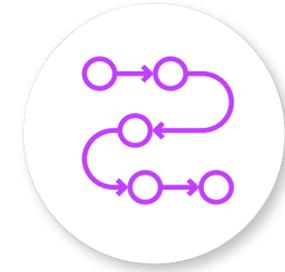
Une idée de produit à confronter à votre marché, clients et utilisateurs.

- Vous souhaitez matérialiser une proposition de valeur à moindre coût.
- Vous souhaitez tester la désirabilité de votre produit auprès de votre marché cible.
- Vous souhaitez tester la viabilité du modèle d'affaire construit autour de votre produit.



Une stratégie de croissance et une roadmap produit à valider.

- Vous souhaitez rapidement matérialiser une nouvelle fonctionnalité à moindre coût, avec l'assurance de pouvoir rapidement faire marche arrière.
- Vous souhaitez tester au plus tôt l'impact de cette nouvelle fonctionnalité sur vos indicateurs clés.
- Vous souhaitez adosser une solution provisoire à une architecture déjà existante.



Une initiative pour optimiser vos outils internes à explorer.

- Vous souhaitez rapidement proposer des solutions sur étagères et peu coûteuses, aux équipes métiers ayant besoin de s'outiller (CRM, outils de productivité...).
- Vous souhaitez valider le ROI d'un outil destiné à l'interne.
- Vous souhaitez renforcer des convictions sur des scénarios business en arbitrage interne.

Le Low-code / No-code facilite 4 composantes clés des projets d'innovation :

- **Le planning**
- **Le budget**
- **La culture**
- **L'aspect durable**



Problème n°1 : le planning

Réduire le temps alloué à la matérialisation d'un concept, d'une fonctionnalité ou d'un poc pour l'interne

Sans Low-code / No-code

5 mois à 1 an

Cela représente le temps moyen d'implémentation d'une nouvelle application pour la plupart des organisations en fonction de la taille du projet.

Avec Low-Code / No-code

24 heures à quelques mois

Selon la taille des projets, le Low-code / No-code permet de réduire le temps de développement de 50 à 90%*.

À retenir

1

Le Low-code / No-code permet, au moins dans un premier temps, de s'affranchir des systèmes legacy. Il permet de fluidifier les interactions avec les services IT, souvent les seuls à posséder des ressources en développement.

2

Le Low-code / No-code permet d'obtenir plus vite une visualisation des fonctionnalités ou du produit fini en production à partir d'un code plus "minimaliste". Il allège les processus de spécification, de test et de correction de bugs. Ainsi, il réduit le nombre d'aller-retours entre équipe métier / PO / UX et développeurs.

Ce gain de temps permet de dérisquer plus rapidement vos idées en avançant le jalon du "reality check" de quelques mois.

*Source : Lehman, 2018

Problème n°2 : le budget

Réduire le coût associé au développement d'un concept, d'une fonctionnalité ou d'un poc pour l'interne

Sans Low-code / No-code

250€ à 800€* / jour

Coût journalier moyen, selon son expérience, d'un développeur (en France)

Avec Low-Code / No-code

0€ à 500€* / mois

Pour lancer un site web ou une application mobile en No-code.

À retenir

1

Le Low-code / No-code permet de ne pas faire appel ou de réduire fortement le besoin en ressource de développement, notamment externe. Même dans le cas du Low-code, les besoins en production, en mise en place d'environnements, en tests et déploiements sont réduits.

2

Le Low-code / No-code permet de s'appuyer sur des infrastructures plus légères et des outils moins coûteux (licences, maintenance...)

Cette réduction des coûts de développement permet d'associer de plus petits budgets aux phases d'exploration et de validation de vos projets. Vous pouvez ainsi en explorer davantage.

Problème n°3 : la culture

Encourager une culture produit lean et agile

Les pré-requis d'une culture LEAN et le Low-code / No-code



Le besoin des clients / utilisateurs avant tout

Le Low-code / No-code permet de garder les discussions autour de la valeur créée pour le client et l'utilisateur au centre grâce à une réduction du temps passé à discuter de la faisabilité technique.



Matérialiser à moindre coût et tester rapidement

Le Low-code / No-code permet un gain de temps et de coût de développement considérable pour donner corps à un concept.



Savoir pivoter lorsque nécessaire

Le Low-code / No-code permet plus facilement d'opérer des changements UX dans la solution. L'approche modulaire permet de faire appel à des outils compatibles pour combler le manque de certaines fonctionnalités. Il est plus facile d'adapter la solution et d'itérer sur les recherches clients / utilisateurs.

Les pré-requis de L'AGILE et le Low-code / No-code



Une collaboration accrue entre les porteurs métiers et les développeurs

Dans le cas du No-code, le porteur métier peut manipuler l'outil lui-même. Le besoin de collaboration est donc tout simplement supprimé. Dans le cas du Low-code, la solution étant plus accessible et rapide à mettre en place, les discussions sont rapprochées dans le temps et se font plus aisément.



Une livraison rapide et continue de nouvelles fonctionnalités

Le Low-code / No-code est adapté aux cycles de développement courts et aux méthodes de priorisation de l'Agile.



Un certain goût pour la simplicité

L'Agile évite au maximum le gaspillage sur des tâches et développements non "essentiels". Le Low-code / No-code met à disposition du code, un nombre limité de fonctionnalités et templates épurés.

Problème n°4 : garantir une conception durable

Le Low-code / No-code s'intègre à une démarche de "numérique responsable"

1

Des templates épurés, peu d'éléments hébergés...

L'approche du Low-code / No-code vous permet de construire une version pragmatique de votre interface, avec un nombre limité de composants et peu d'éléments hébergés. **Cependant, pour garder une vision durable, il convient de choisir les plateformes se développant dans ce sens** (fournisseurs d'hébergements responsables, templates éco-conçus...).

2

Tester avec le Low-code / No-code au préalable...

Privilégier une approche par le test pour vérifier la valeur de votre solution. L'approche du Low-code / No-code permet de rapidement matérialiser un concept / une solution et le mettre dans les mains des utilisateurs. Combinez-la à des tests rigoureux (interviews, survey, carte de chaleur et autres outils d'analytics...) pour éviter de lancer le développement d'un produit lourd et complexe qui se révélera inutile dans le futur.

3

Un comportement responsable de rigueur...

Une fois que vous avez choisi votre outil Low-code / No-code, **il faudra définir des règles d'utilisation des templates**. En effet, le meilleur moyen de réduire l'impact environnemental de votre interface est de réfléchir à vos parcours utilisateurs, vos fonctionnalités et votre contenu selon ces principes : pertinence et frugalité.

En revanche, le Low-code / No-code n'est pas adapté à tous les projets...



Lorsqu'un projet, avec des enjeux de sécurité forts, est en phase de recherche préliminaire ou doit s'intégrer à une architecture complexe



Avez-vous l'un ou plusieurs des enjeux de sécurité suivants ?

J'ai besoin de gérer les serveurs sur lesquels j'héberge mon application...

Pour les applications No-code, aucun doute, vous n'aurez pas le choix du fournisseur d'hébergement. La plupart du temps, ces outils font appel aux grands fournisseurs américains - AWS, Azure ou encore Google cloud - dont les datacenters sont situés aux Etats-Unis et ainsi soumis au Patriot Act et loi d'extra-territorialité américaine.

Des données hautement sensibles vont transiter ou être stockées dans mon application...

Dans des secteurs d'activités bien particuliers comme par exemple la Banque ou la Santé, les données utilisateurs sont hautement sensibles. Les solutions permettant de faire transiter ou bien de stocker ces données sont soumises à des contraintes SI non compatibles avec les outils du Low-code / No-code.



Les outils Low-code / No-code ne sont pas adaptés à votre projet



Avez-vous une architecture complexe à laquelle il faudra complètement intégrer votre projet ?

Pour matérialiser un concept en phase d'exploration, le Low-code / No-code permet de s'affranchir des systèmes legacy. Cependant, lorsque le projet s'avère exploitable, il faut se poser la question de l'intégration en fonction des besoins marché, utilisateur et interne.



Les outils Low-code / No-code sont conseillés pour vérifier la valeur du projet mais garder en tête que la migration vers des outils "code" sera nécessaire.



Avez-vous une bonne compréhension du marché et des besoins clients / utilisateurs ?

Alors que le Low-code / No-code est clé pour matérialiser une expérience réaliste de bout en bout (prise de rendez-vous, paiement...), il n'est pas nécessaire en phase de recherche du besoin. Un storyboard, des maquettes cliquables via Figma, une brochure faite sur Canvas suffisent amplement dans beaucoup de cas.



Itérez d'abord sur cette phase exploratoire via des prototypes primaires avant de vous lancer...

**Pour recevoir la version intégrale de notre
analyse, écrivez-nous :**



contact@thinkmarket.fr



Retrouvez Thinkmarket sur :



Pour en savoir plus : www.itstimetoshift.consulting