

Développer un applicatif informatique pour du crowdsourcing : Regards croisés des expériences CiTIQUE et Pl@ntNet

Diane LE HÉNAFF,
INRAE - DipSO (pôle SENS) à partir des propos recueillis auprès
d'**Antoine AFFOUARD,**
Inria
et **Julien MARCHAND,**
CPIE Nancy Champenoux

Les regards croisés de deux informaticiens, Julien Marchand pour le projet CiTIQUE et Antoine Affouard pour Pl@ntNet, nous éclairent sur les stratégies adoptées pour mettre en place l'applicatif informatique permettant la collecte des données, auprès des citoyens, pour ces projets respectifs. Leur retour d'expérience nous montre l'importance de bien identifier les besoins à mettre en œuvre à travers des fonctionnalités, les difficultés rencontrées par l'évolution des projets (volume de données, besoin de s'adosser à un système d'information local, etc.). Leurs recommandations dépassent le cadre informatique pour constituer tout un ensemble de ressources à prendre en compte : frameworks et base de données, compétences en interne, ergonomie et veille technologique, sécurité des accès et cadre légal pour la gestion des données personnelles et, enfin, le support utilisateur.

Le crowdsourcing est une démarche de recherche de type participatif dans laquelle les citoyens ou des amateurs participent à la collecte de données. Qui dit collecte de données auprès d'un groupe de participants, dit nécessité d'un outil permettant cet échange de données, d'une interface permettant de renseigner les informations utiles aux scientifiques. Pl@ntNet et CiTIQUE sont deux projets de science participative à forte composante crowdsourcing (voir Focus dédiés dans ce numéro). L'expérience de ces projets, après plusieurs années de collecte de données auprès des citoyens, est particulièrement précieuse pour celles et ceux qui voudraient développer une application pour le crowdsourcing. Nous avons choisi de partager leur retour d'expérience sous la forme de regards croisés des deux responsables techniques des projets cités : Julien Marchand et Antoine Affouard.

Informaticien à son compte depuis 10 ans pour « iouston informatique », Julien Marchand est également chargé de mission au CPIE depuis 4 ans (Centre Permanent d'Initiatives pour l'Environnement) Nancy Champenoux, pour le

programme CiTIQUE. Antoine Affouard est ingénieur en recherche et développement à Inria, dans l'équipe Inria-Soft, détaché dans l'équipe Inria Zenith (Lirmm) et à l'UMR AMAP (Cirad, INRAE, CNRS, IRD, Université de Montpellier). Analyste programmeur spécialisé dans le traitement de données de biodiversité et d'environnement, et responsable de l'architecture et des évolutions logicielles de la plateforme collaborative Pl@ntNet depuis maintenant 10 ans, il intervient notamment sur l'analyse et la conception du système d'information Pl@ntNet, sur les bases de données, les logiciels et applications et sur le développement de fonctionnalités innovantes. Leur retour d'expérience permettra d'éclairer les choix techniques et fonctionnels qui ont permis la mise en place de l'applicatif, les besoins spécifiques et les difficultés rencontrées. Ils nous donneront leurs recommandations pour choisir l'outil, aborderont l'importance de la sécurisation des accès au regard de la législation en cours et, enfin, l'intérêt d'avoir un humain dans cet écosystème technique sous la forme d'un support utilisateur ou d'un community manager¹.

¹ Un community manager est un animateur de communautés utilisant les outils du web. Son objectif peut être de fédérer une communauté autour d'un objectif, d'un savoir-faire ou autre.

L'importance de bien appréhender les besoins fonctionnels du projet

Chaque projet a un certain nombre de besoins dont certains peuvent être spécifiques. L'objectif d'un applicatif technique est de permettre, avec des fonctionnalités, de couvrir ces besoins. Dans les projets de crowdsourcing, nous retrouvons beaucoup de besoins autour de la collecte de données : fonction de géolocalisation, permettre à l'utilisateur de décrire ou d'envoyer une image. En effet, dans les expériences rencontrées au sein des projets CiTIQUE et PI@ntNet décrites ci-dessous, il est intéressant de constater le besoin commun de collecter des informations provenant des utilisateurs (besoin classique), mais également la difficulté de s'interopérer avec une application locale de gestion. Chaque projet peut donc avoir des particularités qui peuvent apporter quelques contraintes.

Le projet CiTIQUE consiste à mieux connaître et comprendre l'écologie des tiques et des maladies qu'elles transmettent. La collecte demandée vise à signaler la piqûre de tique sous la forme d'une géolocalisation et d'un contexte associé, et également à récolter et envoyer la tique impliquée au laboratoire pour analyse. La tique est stockée dans une tiquothèque². Julien Marchand nous explique que le premier besoin se satisfait en offrant aux citoyens un formulaire de renseignement et une fonction de localisation GPS à remplir via une application smartphone nommée « Signalement Tique » (application gratuite disponible sur Apple Store et Google Play). Le second besoin de croiser les signalements avec les analyses de tiques sera satisfait par une interopérabilité de l'applicatif de collecte avec l'outil existant, permettant de suivre, stocker et tracer le cycle de la tique reçue en laboratoire. Cette interopérabilité n'avait pas pu être mise en place avec le premier applicatif de collecte déployé. En effet, le ministère de la santé, au travers du plan Lyme mis en place en septembre 2016, était prêt à financer le développement avec la condition de sortir une application mobile de signalement en un temps record. Un développement spécifique aurait alors pris trop de temps, et l'équipe CiTIQUE s'est ainsi penchée sur des applicatifs clés en main qui ne nécessitent que quelques paramétrages pour être opérationnels. Après hésitation avec une application suisse, trop onéreuse, le programme CiTIQUE s'est finalement tourné vers la plateforme e-phytia³, portée et développée par une équipe INRAE de l'UMR SAVE de Bordeaux. Grâce

aux efforts de cette équipe, l'application « Signalement Tique », basée sur e-phytia et paramétrée selon les besoins de CiTIQUE, a été mise en place dans le temps record imposé par le ministère, et a permis de collecter des données en masse, sur toute la France, dès le mois de juillet 2017. Si le choix d'e-phytia a permis de couvrir les besoins essentiels de collecte, il ne permettait pas, en revanche, de s'adosser au système de gestion des tiques piqueuses envoyées par les citoyens et analysées au laboratoire Tous Chercheurs (voir Focus projet CiTIQUE). Les tiques ne sont pas toutes analysées immédiatement, aussi, les chercheurs du laboratoire doivent stocker et gérer une banque de 3 contenus en croissance très rapide et avec plusieurs milliers de données : les signalements, les échantillons de tiques et les résultats associés. Il a donc été nécessaire de revoir l'applicatif, afin de pouvoir lier les signalements reçus à la tiquothèque.

Ainsi, le fait de devoir jongler avec les exigences de commanditaires et financeurs, qui investissent souvent sur du court terme, amène à devoir faire un choix d'outil clé en main. Ces outils clés en main sont très pratiques et souvent peu onéreux, car le coût de développement est déjà en partie absorbé. Mais Julien Marchand nous alerte sur la leçon à tirer de cette expérience : ce coût économisé au démarrage se répercute finalement, au fil du temps, sur une gestion manuelle nécessaire pour pallier la rupture dans le flux d'informations entre le numérique et la gestion sur un autre outil. Le projet PI@ntNet est similaire, de ce point de vue de l'interopérabilité, car la collecte massive de photos de plantes auprès des utilisateurs doit pouvoir être couplée à des informations (validation, discussion) provenant d'un réseau d'experts botanistes organisé par l'association TelaBotanica. L'application PI@ntNet a donc besoin à la fois d'interagir avec les utilisateurs, mais aussi avec des experts pour les identifications des plantes et des nouvelles espèces. Ce réseau d'experts utilise des outils propres avec lesquels l'application PI@ntNet doit donc s'interfacer.

Idéalement, il est recommandé d'identifier ce genre de besoins dès le démarrage du projet pour éviter de devoir refondre l'application plus tard. « Il existe aussi des besoins qui émergent au fil du temps », nous rappelle Antoine. Pour exemple, le besoin de traduction de l'interface PI@ntNet qui a démarré uniquement en français et qui est maintenant traduite en 22 langues. Le succès de PI@ntNet a pro-

² La tiquothèque du projet CiTIQUE est un lieu de stockage physique associé une base de données dont l'accès est réservée aux gestionnaires du projet (www.tiquotheque.fr).

³ Le portail INRAE e-phytia héberge plusieurs applications en santé des plantes permettant notamment d'identifier, de mieux connaître les maladies et ravageurs de diverses plantes cultivées, de choisir des méthodes de protections des plantes pertinentes, et de réaliser de l'épidémiologie. Pour en savoir plus : <http://ephytia.inra.fr>

gressé au fil des traductions de l'application, et les traductions ont permis de capter de nouveaux utilisateurs. C'est un cercle vertueux qui s'est mis en place. Lorsqu'un utilisateur fait part de son intérêt de voir l'application traduite dans une nouvelle langue, l'équipe Pl@ntNet lui propose l'accès aux fichiers de traduction des applications, sites et fiches descriptives. Le côté participatif est également important, au-delà de la production et de la révision des données.

L'applicatif technique : nécessité et points de vigilance

Un projet de crowdsourcing a besoin de collecter des données. Parfois, les besoins sont des données massives que tout citoyen peut apporter, des renseignements textuels aux images ou un besoin de signalement géolocalisé, c'est ce cas qui est abordé ici avec les exemples de Pl@ntNet et CiTIQUE. Mais, certains projets visent à collecter des données plus expertes, comme le cas du projet FLEGME (coordonné scientifiquement par le département MICA) qui consiste à collecter le savoir-faire des citoyens dans la fermentation des légumes. Ces données expertes sont souvent disponibles en plus faible quantité, ce qui ne nécessite pas forcément de mettre en place une infrastructure technique de collecte via une application smartphone.

Tous les développements de l'applicatif technique pour Pl@ntNet ont été faits en interne. Antoine Affouard explique que ce choix a été fait afin de pouvoir répondre rapidement aux nouveaux besoins émergents. En même temps, la large utilisation de Pl@ntNet oblige les sites web et applications mobiles à être stables. Le choix des technologies a donc été fait pour répondre à cette contrainte tout en tenant compte des infrastructures à disposition et des compétences des programmeurs impliqués dans le projet. Les développements reposent sur des frameworks⁴ "classiques" qui ont l'avantage d'être maîtrisés par un grand nombre de développeurs, d'être compatibles avec la majorité des navigateurs Web, et qui permettent de créer rapidement des interfaces qui répondent aux besoins émergents. La majorité des éléments techniques qui constituent l'application Pl@ntNet sont actuellement hébergés sur des serveurs virtuels du Cirad. Antoine nous explique que l'applicatif de Pl@ntNet est constitué de 3 "sous-ensembles" techniques importants :

- Le moteur d'identification est développé en interne et basé sur les travaux de recherche les plus récents en technologies d'intelligence artificielle appliquées à la biodiversité, avec des problématiques d'apprentissage et de classification sur des jeux de données de plus en plus importants (deep learning) et de plus en plus complexes (milieux tropicaux, herbiers). Ce sont les bonnes performances de cet outil, co-développé avec des chercheurs Inria du projet, qui permettent d'attirer et de fidéliser un grand nombre d'utilisateurs et donc de collecter un volume croissant de données.
- Les interfaces d'administration et les API internes au projet : ce sont des botanistes et écologues (Cirad, IRD, INRAE) qui participent à maintenir la cohérence des données botaniques utilisées et diffusées sur la plateforme, et qui doivent donc pouvoir contrôler la qualité des données. Pour cela, des interfaces d'administration ont été développées pour faciliter l'exploration et la curation des données. Ces outils sont également très importants car ils permettent de proposer des informations de qualité et, donc, de renvoyer une image positive aussi bien au grand public qu'aux partenaires scientifiques du projet. Cela permet aussi de contrôler plus finement les données transmises à l'apprentissage du moteur d'identification et, donc, d'assurer une constante progression des performances en termes de qualité et de quantité de données gérées. Pour les travaux de recherche en écologie, des API internes au projet permettent de récupérer des jeux de données en filtrant sur un certain nombre de critères. L'objectif de ces API est de faciliter les échanges de données en se basant sur des standards (données facilement interprétables et dans des formats compatibles avec une large gamme d'outils) et de minimiser les extractions directes en base de données et, donc, de minimiser les interventions d'informaticiens.
- Les interfaces publiques et les API publiques : ces interfaces sont destinées à être utilisées par le grand public, aussi bien par des novices en botanique que par des passionnés, des professionnels de différents domaines ou des experts. Elles doivent donc être suffisamment simples d'utilisation pour être accessibles au plus grand nombre, mais proposer du contenu d'intérêt pour tous les niveaux d'expertise (noms vernaculaires vs. nom binomial, par exemple). Au début du projet, il n'y avait qu'un site web qui permettait de soumettre des images au moteur d'identification. Par la suite, avec la démocratisation de l'utilisation des smartphones et l'implication de développeurs avec

⁴ Un framework facilite le travail des développeurs informatiques en leur offrant une architecture "prête à l'emploi" qui est enrichie par les développements des autres.

⁵ API est une interface de programmation d'application. C'est une solution informatique qui permet à des applications de communiquer entre elles et d'échanger des données ou des services.

des compétences variées, des applications mobiles ont également été développées (sur iOS dans un premier temps, puis sur Android). Ces applications mobiles sont particulièrement pertinentes dans le cadre de PI@ntNet, car elles permettent l'identification des plantes directement sur le lieu d'observation et ont donc participé au succès du projet. Les données ainsi collectées sont géolocalisées et donc très utiles dans les travaux de modélisation d'aires de répartition par exemple. Les API publiques permettent de collecter et de diffuser des données à l'ensemble des utilisateurs de PI@ntNet. C'est la porte d'entrée de toutes les données aujourd'hui disponibles.

Pour CiTIQUE, la nouvelle plateforme, déployée en 2020, s'appuie sur un logiciel libre open source : Dolibarr. Dolibarr est un logiciel de gestion d'entreprise et d'association porté par une communauté de développeurs et animé par une association française. Même si le logiciel ne répond pas strictement aux besoins spécifiques de CiTIQUE, il dispose de tout le code nécessaire pour permettre rapidement de développer l'applicatif qui était souhaité. Dolibarr a donc été utilisé comme un framework, c'est-à-dire une structure logiciel ou un ensemble de composants sur lesquels sont développées les fonctionnalités spécifiques. Les frameworks évitent d'écrire des fonctions simples mais systématiquement nécessaires, comme par exemple : ajouter, modifier, supprimer une donnée à la base de données, ce que les informaticiens appellent dans leur jargon le CRUD (pour : Create Read Update Delete). Dolibarr disposait de tout le code de base nécessaire pour permettre de développer rapidement un outil très spécifique et de gagner du temps. Comme indiqué parmi les besoins importants du projet, cette plateforme développée est constituée d'un système de gestion de contenus spécifiques en lien avec notre tiquothèque : les signalements, les envois, les échantillons et les résultats des analyses. Le tout est accessible en ligne depuis le site web de la tiquothèque. Pour que les internautes puissent alimenter cette base de données, un formulaire web, un formulaire papier et deux applications mobiles (sous iOS Apple et Android) sont disponibles. Cette application, déployée en 2020, apporte énormément de nouveautés, parmi lesquelles :

- Des notifications 'push' que l'administrateur de la tiquothèque déclenche manuellement ou de façon programmée et qui apparaissent sur tous les smartphones où l'application est installée (21 000 utilisateurs à ce jour).
- Le fait pour un utilisateur d'avoir facilement accès à la

liste de ses signalements et à leurs numéros respectifs demandés préalablement à l'envoi des tiques piqueuses⁶. Ce numéro nous est nécessaire pour faire le lien entre les données saisies par l'utilisateur lors du signalement et l'échantillon envoyé.

- La possibilité de créer des profils d'animaux et d'utilisateurs, de façon à éviter à l'utilisateur les saisies répétitives de données et/ou de façon à personnaliser les formulaires renseignés. Par exemple ? un utilisateur peut avoir dans son application : 1 profil pour sa fille, 1 profil pour son fils, 1 profil pour son chat et un profil pour lui, en tant que garde ONF par exemple.
- La possibilité de signaler des piqûres sur son smartphone, même lorsqu'il n'y a pas de couverture réseau. Dans ce cas, l'application télétransmettra les signalements à la tiquothèque lorsque le réseau fonctionnera à nouveau.

Pour les administrateurs et gestionnaires travaillant pour le programme CiTIQUE, l'articulation entre l'application mobile et la tiquothèque apporte des fonctionnalités intéressantes avec cette nouvelle architecture, parmi lesquelles :

- La possibilité de forcer la mise à jour à distance de l'application, pour forcer la mise à jour des formulaires et éviter les disparités de saisie, tout en laissant la possibilité à l'équipe CiTIQUE de faire évoluer ces formulaires;
- Le renseignement automatique, dans le formulaire de saisie, de la région, du département et de la ville depuis les coordonnées GPS fournies par l'application;
- La possibilité de choisir de crypter en base des informations personnelles, de façon à rester conforme au Règlement Général sur la Protection des Données (RGPD);
- Des données transmises sécurisées grâce au protocole SSL qui crypte les données en permanence;
- La possibilité d'afficher de façon conditionnelle certains champs du formulaire;
- La disponibilité d'une API (Interface de Programmation pour accéder en lecture et/ou en écriture aux données) pour faciliter le partage des données et le travail en réseau avec d'autres chercheurs et d'autres laboratoires;
- La possibilité de dupliquer des champs de formulaire pour mettre en place facilement et rapidement des formulaires particuliers;
- La possibilité d'envoyer en masse, et de façon semi-automatique, des accusés de réception des tiques;

⁶ A chaque signalement via l'application smartphone, un numéro à 6 chiffres est donné à l'utilisateur, numéro qui servira à tracer et reconnaître la tique piqueuse envoyée.

- Une vue générale sur l'ensemble des données du programme;
- une facilité et un gain de temps pour ressortir des données par territoire et rester ainsi en contact avec les partenaires locaux et de proximité qui participent au programme.

Retour d'expérience sur les difficultés rencontrées

Le projet CiTIQUE avait besoin de faire évoluer son applicatif afin de couvrir des besoins comme la possibilité de récupérer les signalements dans des zones non couvertes par le réseau (mode non connecté) et faciliter l'échange d'information avec la tiquothèque. Ce besoin de faire évoluer les fonctionnalités s'est traduit par l'appropriation en interne d'un framework. Le déploiement du nouvel applicatif en 2020 a donc permis de faire un saut fonctionnel nécessaire. Il en a été de même pour Pl@ntNet qui a également fait le choix de maîtriser en interne les outils utilisés. La principale difficulté pour Pl@ntNet est de gérer un volume important et croissant de données collectées, (qui double chaque année !). Les infrastructures auxquelles les gestionnaires et administrateurs de Pl@ntNet avaient accès avant la migration ne pouvaient pas héberger plus de 2 To de données par serveur. Il a fallu changer de système de gestion de base de données. La base de données CouchDB (NoSQL), qui était alors utilisée, est très gourmande en espace disque, ce qui rendait difficile l'implémentation de nouvelles fonctionnalités (lenteur de fonctionnement constatée). Le choix a donc été fait d'utiliser la base de données ArangoDB (toujours NoSQL) qui répondait à l'ensemble des besoins tout en divisant l'espace disque occupé par 20 (3 To (2 VM de 2 To) -> 150 Go) et facilitait la manipulation des données. Cette solution permet également de distribuer les données sur un cluster de serveurs (regroupement de plusieurs serveurs indépendants ne formant qu'un seul système global), afin de répondre aux problématiques de croissance du volume de données stockées. En outre, certaines API, ayant besoin d'être modernisées, ont également été remplacées.

Une autre difficulté, rencontrée dans cette « réinformatisation » des infrastructures Pl@ntNet, était due à la compétence nécessaire pour administrer, maintenir et faire évoluer les outils techniques. L'équipe en charge de ces questions n'avait plus de développeur compétent sur les outils qui permettaient de maintenir et développer l'application smartphone iOS (pour Apple). Ainsi, le choix d'utiliser un framework qui crée des applications Android et iOS à partir d'un seul et unique code a permis de contourner

cette difficulté de compétence spécifique.

Des conseils à donner à des collègues qui voudraient développer un applicatif pour du crowdsourcing

Grâce à l'expérience CiTIQUE, Julien Marchand mentionne 11 points importants à prendre en compte avant de se lancer dans le développement d'un applicatif :

1. L'existant

Il est nécessaire d'investir un certain temps avant même le démarrage du projet pour voir ce qui existe déjà et ce qu'il se fait du point de vue technique et fonctionnel. Ce travail consiste à faire de la veille.

2. Le langage informatique

Éviter la course à la technologie « tape à l'œil » ! L'informatique évolue très vite, et certains frameworks derniers cris n'existeront peut-être plus d'ici 3 ans. Le recours à des technologies simples et éprouvées est souvent plus prudent selon Julien, qui illustre cela en rappelant que « tout le système bancaire mondial repose toujours sur du cobol, un langage des années 60, mais fiable et éprouvé ». Dans le cas de CiTIQUE, c'est le langage PHP, avec de nombreuses classes pour écrire rapidement l'application, qui a été retenu. « Ce langage n'est sans doute pas parfait, mais il est maintenu, il continue à évoluer, il est éprouvé, et tous les développeurs web ou presque le maîtrisent »,

3. La base de données

Au-delà du choix du langage utilisé, il y a aussi la technologie côté serveur. Aujourd'hui, il est possible de recourir à différents systèmes de base de données (SQL/MySQL / postgresSQL / CouchDB / MongoDB, etc.) dont certains sont fournis d'office par tout hébergeur, tandis que d'autres nécessitent des infrastructures plus complexes. Dolibarr peut tourner sous MySQL, MariaDB et PostgreSQL. L'équipe CiTIQUE est restée sur MariaDB, un système de gestion de base de données simple, puisque l'équipe ne dispose pas dans l'immédiat (ni prévu dans le futur) de personne dédiée à l'administration système.

4. La cohérence entre la technique, les compétences de l'équipe et le budget

CiTIQUE ne dispose ni d'un budget important pour l'infrastructure serveur ni des compétences d'un administrateur système, mais l'équipe qui a développé l'applicatif à partir de Dolibarr et MariaDB était parfaitement à l'aise avec ces outils. Pendant leur benchmark de l'existant, ils avaient repéré certains logiciels intéressants comme opendatakit.org, mais qui nécessitaient un réapprentissage complet pour le développeur en charge du projet, et donc une moins bonne visibilité

de la faisabilité technique pour le chef de projet.

5. La simplicité de la demande et l'ergonomie

Il faut également veiller à la quantité et à la nature des informations scientifiques demandées aux citoyens, qui peuvent être très claires pour le spécialiste, mais plus critiques pour les non-initiés. Il faut réussir à récolter les données dont les scientifiques ont besoin sans pour autant « en demander trop » aux citoyens. Un équilibre à trouver ! L'outil qui sert à récolter les données doit être intuitif, simple, captivant. Sinon, il y a le risque de perdre beaucoup d'utilisateurs potentiels avant même qu'ils n'utilisent effectivement l'application. Le fait de changer d'application en cours de route a permis de retravailler complètement l'ergonomie de l'application mobile ainsi que son circuit d'utilisation. Pour cela, l'équipe CiTIQUE a été accompagnée par Thibaut Marquis, designer spécialisé en expérience utilisateur. « Cela a été payant : nous avons récolté autant de signalement sur la saison 2020 en 6 mois avec la nouvelle application que sur les 3 années précédentes du programme. Il ne faut donc pas sous-estimer ce poste ! ».

6. Le choix de l'open source

Selon Antoine, choisir des technologies ouvertes est une façon de concorder avec l'esprit d'un projet de science participative. Par ailleurs, c'est sécurisant pour le projet. En effet, un logiciel open source permet à tout le monde de lire et modifier le code. Cela sous-entend donc que n'importe quel informaticien peut proposer une correction ou une évolution de votre code. « Cela n'a l'air de rien mais ça change tout, car si demain, pour une raison ou une autre, vous deviez changer de prestataire de développement, d'hébergement, vous êtes libre de faire appel à quelqu'un d'autre. Rappelons-nous la faille de sécurité « utilman.exe » connue de tous les informaticiens depuis windows 98. » Cette faille de sécurité permet de contourner assez facilement le mot de passe de n'importe quel ordinateur windows. « Cette faille est bien sûre connue et signalée à Microsoft, mais logiciel propriétaire oblige, seul un technicien de chez Microsoft est en droit de corriger cette faille. Résultat, elle est toujours présente dans windows 10, pour le plus grand bonheur des dépanneurs informatiques dont les clients ont oublié leur mot de passe. »

7. Jouer et faire de la science

Pour maintenir la motivation des participants, il s'agit, c'est un grand classique, de récompenser par des points, des badges, des niveaux ou toute autre distinction des utilisateurs selon leur participation. Selon Antoine, l'application INPN Espèces fait très bien cela. Chaque fois qu'une espèce animale ou végétale est si-

gnalée sur l'application, des experts attribuent un certain nombre de points en fonction de certains critères, comme la rareté, l'exactitude de la reconnaissance de l'espèce, la photo, la faible fréquence de données dans un lieu donné, etc. « Il y a une mise en réseau des participants; ainsi, il est possible de s'apercevoir que certaines personnes que l'on pourrait connaître, pas du tout naturalistes à la base, participent avec plaisir, non pour la science ou par intérêt naturaliste, mais juste pour marquer des points ». Cet élément de ludification (ou gamification) est un des axes sur lesquels Julien aimerait travailler pour faire avancer l'application, même si ce n'est pas prioritaire pour l'instant.

8. L'information

S'il faut « faire jouer », il faut aussi renseigner l'utilisateur. Il s'agit d'une logique d'échange mutuel (donner-avoir). « L'utilisateur nous fait parvenir des données qui nous servent à faire avancer la recherche, mais de notre côté nous devons fournir des retours sur ces données et des informations, par exemple, sur la prévention des piqûres des tiques. »

9. Maximiser le côté participatif

Dans le cas de collecte de données par des personnes non expertes, il est important d'effectuer une validation sur la qualité de ce qui a été collecté. L'application espagnole Mosquito Alert recueille des signalements de présence pour les moustiques tigres. Les citoyens peuvent prendre en photo ce qu'ils identifient comme des moustiques tigres. Mais, avant que l'information ne parvienne aux scientifiques du projet, les données sont présentées via l'application à un panel d'utilisateurs avancés. Ces derniers font un premier tri des données pour réduire le taux de fausses identifications et ainsi mieux qualifier les données. Cela permet d'organiser le tri des données de façon participative. Dans le cadre de CiTIQUE, cette vérification est toutefois difficile à mettre en place, du fait de la difficulté de déterminer les espèces de tique d'après photos.

10. « Le papier n'est pas mort ! »

C'est ainsi que s'exclame Julien qui nous invite à considérer que tout le monde ne dispose pas de smartphone ou d'ordinateur. Pour une raison technique, la meilleure application du monde peut dysfonctionner (et à l'heure où nous écrivons cet article, ce ne sont pas les hospitaliers de Dax, dont tout le système informatique a été piraté et qui n'ont plus accès à aucune donnée, qui viendront nous contredire). Dans de nombreux autres cas de figure (facilité de saisie, couverture réseau, éloignement, collecte spécifique au cours de manifestation, etc.), le papier est utile. Dans le cas de CiTIQUE, beaucoup de signalements sont encore reçus sur formulaires papier. Et dans ce type de collecte sous

forme papier, ils font appel à d'autres citoyens volontaires pour aider à la saisie de ces informations dans la base de données de signalements.

11. « Documentation, documentation »

C'est le dernier conseil de Julien Marchand : « Documentez votre applicatif. D'une façon ou d'une autre, prenez le temps d'écrire de la documentation technique, afin de faciliter l'évolution de votre applicatif et ainsi capitaliser l'utilisation que vous faites de votre application. J'en profite pour signaler ici l'excellent Docsify qui permet de générer et maintenir une documentation technique en ligne. L'utilisation de Markdown (un langage syntaxique facile à lire et à écrire pour l'écriture de pages web) permet de se concentrer sur la rédaction tout en obtenant un résultat propre et fonctionnel. »

Pour Pl@ntNet, le point important se situe surtout autour de la motivation de l'internaute, c'est-à-dire proposer un service qui ait un attrait autre que la seule participation à un projet de sciences participatives. Comme le dit Antoine Affouard : « ces passionnés de science seront de très bons contributeurs, mais ils ne seront hélas pas très nombreux ». Antoine rejoint également Julien sur l'intérêt de certains utilisateurs par le côté compétitif que peut prendre la collecte et la révision de données. Établir un classement peut être la source d'un engagement plus important des plus compétiteurs d'entre eux ! Et surtout : toutes problématiques doivent être prises en compte dès le début du projet. Sur ce dernier point, Antoine recommande de lister les besoins fonctionnels existants, mais également hypothétiques, dès le début du projet, afin de prévoir une architecture logicielle et une architecture de base de données adaptées. En effet, nous avons vu que, face au succès de Pl@ntNet, la problématique de la volumétrie a nécessité de migrer vers une nouvelle architecture en cours de projet. Ainsi, le choix des frameworks et des outils informatiques doit se faire, selon Antoine, à travers plusieurs critères :

- la réponse aux problématiques du projet (problématiques fonctionnelles et techniques),
- la stabilité et la maintenance des outils retenus : fréquence de publication des mises à jour de sécurité, compatibilité entre les versions, durée du support pour les anciennes versions, réactivité lors du signalement de bugs ou de demande d'informations techniques, ...
- les affinités et compétences des programmeurs du projet.

Retour d'expérience et conseil sur la question de la sécurisation des accès

La sécurisation des échanges et de l'accès aux données est importante à deux niveaux : d'un côté, l'architecture technique doit être pensée pour réduire les intrusions malveillantes et, de l'autre, il s'agit de protéger les données et les informations enregistrées dans la base de données.

Sur la question des intrusions, Antoine Affouard recommande que les fonctionnalités critiques d'un projet, comme le blocage de comptes utilisateurs, la manipulation des données taxonomiques, l'ajout d'administrateurs, ne soient disponibles que sur les interfaces d'administration et sur les API destinées à exécuter ces opérations. Ainsi, les API publiques permettent uniquement de réaliser des opérations qui ne peuvent pas avoir d'incidence sur l'ensemble de la plateforme (création d'observations, requêtes d'identification, liste d'espèces...). Malgré ces précautions élémentaires, Pl@ntNet a souvent eu des problèmes liés à la lecture massive des données par des scripts ou logiciels ne faisant pas partie du projet. Pour pallier cela, l'équipe Pl@ntNet a mis en place des quotas limitants qui n'autorisent qu'un certain nombre d'actions de lecture ou écriture par jour et par utilisateur (authentifié ou non). En plus de cela, l'équipe a déployé une API spécifique qui donne accès à certaines des données ainsi qu'au système d'identification. Cela permet d'orienter ces connexions vers un point spécifique qui est plus simple à contrôler, car hors du flux des utilisateurs classiques. Enfin, pour tous les échanges de données entre les différents services et outils internes, des serveurs virtuels sont utilisés. Ces serveurs virtuels ne sont pas accessibles depuis Internet et ne peuvent donc pas être affectés par ces éventuelles "attaques".

Sur la question de la protection des données, Julien Marchand rappelle que le règlement général sur la protection des données (RGPD) a changé le paradigme sur la sécurité des informations personnelles. Désormais, l'éditeur est responsable des données et devra prouver, en cas de fuite, qu'il avait mis en place tout ce qu'il était possible de faire. Le cryptage et la protection des données sont donc nécessaires dans la base de données. Les connexions devront, elles aussi, être cryptées (le fameux https://, ou SSL pour les techniciens). L'accès à la base de données devra être limité aux seuls administrateurs techniques de l'outil et déclaré dans les documents RGPD de l'application. Julien nous informe de la possibilité de soumettre un applicatif à un test de sécurité. C'est le principe du bug bounty : offrir une récompense sur une plateforme spé-

cialisée à qui arrivera à pirater votre système. Il précise : « des white hackers (des hackers qui hackent pour la bonne cause) et des spécialistes de la sécurité essayeront alors de trouver une faille, puis vous la signalent pour empocher la récompense. Comme tout concours, plus la récompense est belle, plus les participants se donnent du mal. Mais cela vous permet de traquer des failles avant qu'un pirate mal intentionné ne s'y attèle ».

Importance d'un support utilisateur ou d'un community manager

Dans les projets de sciences participatives, les interactions avec les citoyens, amateurs et non amateurs, sont la clé du succès. Quand une interface applicative est utilisée pour collecter des données, cela induit des difficultés potentielles qui nécessitent une interaction avec un humain. Il s'agira, par exemple, de questions sur l'utilisation de l'outil ou certaines incompréhensions pour envoyer les données. Il est souvent primordial de réduire les sollicitations, car la masse de réponses à traiter peut être très vite ingérable, notamment lors des périodes où les tiques sont actives dans le cas de CiTIQUE. La mise en place d'une Foire Aux Questions (F.A.Q.) apportant des réponses claires et précises sur les questions fréquemment rencontrées (à tenir à jour) est recommandée. Julien Marchand mentionne qu'il est aussi possible de créer des réponses-modèles dans les formulaires. Même avec cette stratégie, le retour d'expérience de CiTIQUE témoigne d'une forte sollicitation. C'est donc un poste budgétaire ressource à ne pas oublier dans le financement, car ne pas apporter de réponses contribuera vite au désintérêt des participants.

Antoine Affouard pense également que le rôle d'un community manager ou d'un support utilisateur est important dès que la communauté d'utilisateurs devient assez conséquente. Il ajoute que c'est cette personne qui va interagir de manière privilégiée avec la communauté en utilisant différents canaux (et pas seulement les emails). Les échanges avec les utilisateurs/participants permettent à l'équipe de recherche de connaître les attentes, les difficultés, les incompréhensions, les satisfactions des utilisateurs et donc d'adapter les développements futurs. Un community manager permet aussi d'informer la communauté, d'événements organisés, de communiquer sur des bonnes pratiques à adopter dans l'utilisation des logiciels et de ne pas laisser de questions sans réponse (ce qui peut être frustrant ou laisser penser que le projet n'est pas vivant).

Des réponses permettent donc de rendre le projet vivant et de créer une dynamique favorisant la motivation. Le rôle du community manager est donc important et ne doit pas être sous-estimé. Julien Marchand explique que CiTIQUE reçoit énormément de témoignages spontanés, qui n'attendent pas forcément de réponse technique, mais qui attendent une réponse empathique et de sollicitude. Ces réponses peuvent influencer la participation de beaucoup d'utilisateurs. En effet, si un utilisateur reçoit une réponse personnalisée et qui met l'accent sur l'importance de sa participation, il motivera son entourage à participer au projet, ce qui agrandit le panel des utilisateurs, créant ainsi un cercle vertueux.

Conclusion

On notera l'importance de prendre du temps en amont pour étudier les besoins et ceux qui pourraient apparaître dans différents scénarios d'évolution. Antoine Affouard comme Julien Marchand pointent du doigt l'importance de maîtriser l'architecture en interne avec le choix de l'open source, d'un framework adaptable, d'un langage de programmation éprouvé et pour lequel on trouve facilement des compétences. Les aspects techniques nécessitent donc des ressources (humaines et financières) qu'il ne faut pas négliger. Le choix d'un outil clé en main permet, certes, un développement rapide, mais entraîne des limitations sur l'évolutivité. Si les besoins du projet évoluent de manière spécifique, il faudra alors se réapproprié un outil en interne. C'est en tout cas le retour d'expérience de CiTIQUE et le choix technique de Pl@ntNet, qui sont des projets de long terme. Un outil clé en main peut néanmoins être très utile dans des projets à court terme, tout dépend du besoin propre au projet de crowdsourcing. Une autre ressource importante, qui a été mentionnée, est celle d'un community manager, c'est-à-dire une personne qui répondra aux problèmes techniques des utilisateurs mais également qui, à travers des retours d'expériences, pourra dessiner les futurs besoins d'évolution de la plateforme. Que l'échange soit dans la résolution d'un problème, un remerciement suite à un témoignage, des informations ou une communication sur des événements, cela montre un projet vivant, dynamique qui incite les personnes à participer. La dimension humaine est donc également importante ! ■