



# L'art de la démo

Olivier Ridoux

► **To cite this version:**

| Olivier Ridoux. L'art de la démo. Master. France. 2021. hal-03213321

**HAL Id: hal-03213321**

**<https://hal-univ-rennes1.archives-ouvertes.fr/hal-03213321>**

Submitted on 30 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

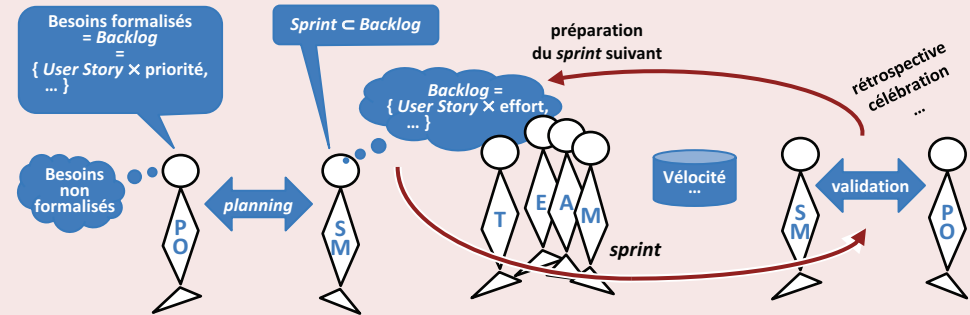
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Prérequis :** agilité et communication - lire, apprendre, comprendre.

**Pliage :** recto visible (autre face), traits gris rentrants, traits rouges saillants. Découper selon le trait rouge entre les deux ● de l'autre face, puis achever le pliage.

### 1 Le cycle de vie agile



Le client (*product owner* ou **PO**) présente ses besoins sous la forme de *user stories* (des cas d'usage) agrémentées d'une indication de **priorité**. L'ensemble constitue le **backlog**. Les *user stories* s'expriment le plus souvent en termes client (quand on fait ceci, on obtient cela) qui peuvent servir de cas de test. Le **scrum master** (SM) et son équipe de développement évaluent l'**effort** demandé par chaque *user story*.

Les priorités s'expriment en valeurs **ordonnables** (1, 2, etc.), mais pas additives. C-à-d. pas dans l'esprit d'exprimer que 2 est deux fois plus prioritaire que 1. L'effort s'exprime en **point d'effort**. Ce sont des valeurs **ordonnables** et **additives**; 4 représente un effort double de 2. On ne cherche pas à faire une estimation très fine, mais plutôt à dégager des classes. Pour cela on adopte souvent des échelles standardisées qui obligent à écarter les estimations, comme l'échelle de Fibonacci (1, 2, 3, 5, 8, etc.)

Les *user stories*, priorités et points efforts sont les données de base de la planification qui permet de constituer des **sprints** de durée fixée à l'avance, ex. 2 semaines, et qui satisfont au mieux le PO (en privilégiant les *user stories* de plus grandes priorités) tout en étant compatible avec la capacité d'effort de l'équipe ( $\Sigma \text{ effort} \leq \text{capacité}$ ).

À la fin d'un *sprint*, le PO valide, ou non, les *user stories*, exprime un retour sur les services produits, et peut modifier le **backlog**. La validation se fait essentiellement via des **demos** qui réalisent les cas d'usage attendus. Puis, tout le monde planifie un nouveau *sprint*, et on recommence.

En permanence, la **vélocité** de l'équipe est mesurée en terme de points d'effort par *sprint*. Sans valeur absolue ces mesures permettent à l'équipe d'affiner son estimation de l'effort demandé par chaque *user story*.

Ce modèle itératif a entre autre la vertu de permettre une amélioration continue, mais assez rapide, via des expériences (ex. l'évaluation de l'effort et la planification) reproduites assez souvent. Plus généralement, l'agilité permet que les erreurs des uns ou des autres (ex. expression des besoins, analyse, programmation, estimation des coûts) soient détectées et corrigées le plus tôt possible.

### N'oubliez jamais !

Même dans le cadre d'une activité technique, la communication exige **rigueur** et **correction** dans l'emploi de la langue : orthographe, grammaire, argumentation. Mais aussi dans le choix des circonstances utilisées pour illustrer le propos dans une démonstration (contre-exemple vécu : des informaticiens, spécialistes de la simulation de l'activité du cerveau, qui expliquent « **On s'amuse à mettre une tumeur, et on regarde ce qui se passe !** »). Choisir des circonstances tragiques ou très sensibles permet sans doute de capter l'intérêt, mais cela suscitera le rejet quand elles sont traitées avec désinvolture. En plus, cela expose à ce que l'auditeur naïf ou très concerné ne voient qu'elles.

### 1 Le diaporama qui aide

Il est souvent utile d'introduire et de conclure une démo par un **diaporama** (ne jamais dire un **power-point** ! Ne pas confondre l'objet avec l'outil qui l'a fabriqué.). L'objectif est d'aider le présentateur à ne rien oublier de ce qu'il a à dire, et l'auditeur à retenir ce qu'il faut retenir.

Surtout ne pas suivre les recommandations stéréotypées du genre **pas de phrase, seulement des mots clés** ou **pas plus de 6 points par diapo, et pas plus de 6 mots par point**. S'en tenir aux objectifs :

- **Aider le présentateur** : un bon moyen de perdre son auditoire est de se rendre compte à la diapo 15 qu'on a oublié de dire quelque chose à la diapo 5. Pour éviter cela, la scénarisation permet d'identifier les points importants, et les répétitions permettent d'identifier les points qu'on oublie tout le temps. Il vaut mieux un diaporama un peu verbeux mais qui aide, qu'un diaporama qui entretient la confusion.
- **Aider l'auditeur** : il a ses faiblesses, ex. des moments d'inattention, des lacunes, des présupposés erronés. Les hypothèses et les conclusions de la présentation doivent être affichées clairement. Dans tous les cas, c'est la faute du présentateur si l'auditeur ne retient pas la bonne chose.
- **Aider les deux** : il est vain de présenter sans support écrit des noms propres (personnes, lieu, société, produit, marque, ...), sauf si ils sont déjà très connus. Pareillement, pour des quantités, dates, etc. Tout cela doit être écrit si on veut que ce soit retenu. Et si il n'est pas important que ce soit retenu, pourquoi en parler ? Sans support écrit, une phrase comme **En 19xx, M. X créait la société S, dont le siège est à T, qui a maintenant Y salariés dans Z pays, pour un chiffre d'affaire de C** est du gâchis de temps, soit que ces données ne seront pas mémorisées, soit qu'elles sont déjà connues.

### 2 Le diaporama qui nuit

Un excellent moyen pour un diaporama de nuire à un orateur et de ne pas dire la même chose que lui, ou pire de le contredire.

Nous sommes ainsi faits que généralement le canal écrit nous demande plus d'attention que le canal oral, mais est plus rapide. En conséquence, le diaporama qui nuit le fait parce qu'il prend le dessus sur l'orateur. Cela peut arriver de plusieurs façons :

- Le diaporama présente de l'information qui n'est pas oralisée. C'est le cas typique de diaporamas trop fournis et que l'orateur ne fait que survoler. Inmanquablement, l'auditeur va se lancer dans une course de vitesse avec l'orateur pour déchiffrer le diaporama plus vite que l'orateur ne le survole.
- Le diaporama présente la même information que celle qui est oralisée, mais il la présente autrement. Cela crée un conflit cognitif que l'auditeur résoudra en ignorant l'orateur.
- Le diaporama parle d'autre chose que ce que dit l'orateur. Ou dit le contraire ! Cela semble absurde, mais cela arrive.
- Le diaporama présente exactement la même chose que l'orateur ; en fait, c'est comme si l'orateur le lisait. Dans ce cas, le mieux qu'ait à faire l'auditeur est de le lire aussi.

Pour que le diaporama aide il faut que la relation entre ce qui est dit et ce qui est écrit soit extrêmement simple. Par exemple, que soient écrits les points qui sont développés à l'oral, ou que soient écrites les hypothèses, définitions et conclusions à retenir. Et tout ce qui par nature passe mal à l'oral : orthographe, chiffres, noms propres, formules, etc.

### 3 Le vu et le lu

Lire demande un effort d'attention que ne demande pas le simple fait de voir. Cela explique des recommandations/oukases qui interdisent d'utiliser des phrases dans un diaporama, ou même limitent le nombre de mots par page. Ce sont des recommandations qui deviennent absurdes quand elles font fi du contenu réel de la communication. En effet, il est des contenus qui doivent être lus littéralement : poésie, articles de loi, citations, mais aussi formules, spécifications... Ce sont des données de certaines communications.

D'autres données doivent souvent être écrites : noms propres, de personnes, de localités, de produits, quantités, ..., car sans cela elles se transforment en bouillie sonore, et à condition évidemment que ces données servent dans la suite de la présentation.

Les textes conçus pour être **vus** et ceux pour être **lus** sont souvent entourés de modalités typographiques différentes. Par exemple, les guillemets signalent une citation verbatim, donc un texte pour être lu. Les polices avec empattements (**serif**, en anglais) sont souvent le signe d'un texte pour être lu, alors que les polices sans (**sans serif**) sont celui d'un texte pour être vu ; comparer par exemple la typographie des affiches, pour être vu, et celle des imprimés classiques, pour être lu. Même si l'impression moderne utilise de plus en plus souvent des polices sans serif (par exemple, **calibri** pour ce document) pour des textes à lire.

L'impact des empattements sur la lisibilité est très débattu, des études se prétendant scientifiques démontrant une chose ou son contraire. Il faut d'abord se souvenir du rôle du support (ex. pierre, bois, papier, écran), du procédé de transfert (ex. gravure, peinture, gravure, rayon cathodique), et de la qualité de l'ensemble. Des arguments qui tenaient pour des écrans de mauvaise qualité peuvent ne plus tenir pour des écrans modernes, et encore moins pour du papier électronique. Mais une chose est certaine, avec et sans empattement constitue un contraste visuel important qui permet de distinguer facilement le **vu** et le **lu**.

Une autre chose est certaine, le **vu** et le **lu** sont deux modalités de présentation tout aussi légitime l'une que l'autre. Il faut seulement les utiliser à bon escient.

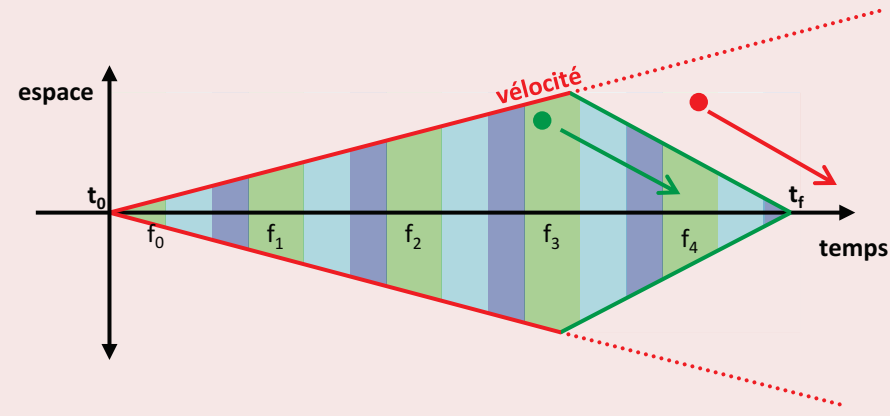
### 2 La démo et l'espace-temps de l'agilité

À partir du moment où on fixe une limite de vitesse pour une progression, on sait qu'il n'est pas possible d'avancer d'une quantité arbitrairement grande en un temps donné. C'est un principe qui vient de la physique, où il a été considérablement renouvelé par la relativité de Einstein. On emploie souvent ce principe sous la forme d'un **cône d'accessibilité** qui s'élargit vers le futur et délimite l'espace qu'on peut atteindre en un temps donné. Par symétrie, on peut concevoir un cône d'accessibilité vers le passé. Il délimite les points de l'espace d'où on peut être venu en un temps donné.

Ce principe s'applique aussi à la gestion du temps dans un projet de développement logiciel. Dans ce cas, l'espace n'est pas celui de la physique, mais celui des fonctionnalités à réaliser. Le plus dur est de connaître la vitesse de développement. Et c'est même une des choses les plus difficiles qui soit à faire, alors que avoir une idée de la vitesse de développement est essentiel pour estimer le coût d'un projet. Les approches agiles à la **Scrum** propose un protocole d'estimation de la vitesse tout au long de la vie d'une équipe.

L'idée est de confronter les estimations de coût en **points d'effort** faites lors de la planification d'un *sprint*, avec la quantité de fonctionnalités, les *user stories*, réalisées à la fin du *sprint*, et mesurée comme la somme des points d'effort des *user stories* validées. On en déduit la **vélocité** de l'équipe, mesurée en point d'effort par durée de *sprint*. La **vélocité** peut être modérée par la **disponibilité** de chacun sur le projet ou pour un *sprint* (ex. pour gérer les périodes de vacances). L'objectif est d'obtenir une estimation toujours meilleure de la **vélocité**.

Une fois cela mis en place on peut imaginer le diagramme espace-temps d'un projet de développement agile. Il alterne des *sprints* qui comprennent chacun un peu d'**analyse**, de **développement** et d'**intégration et validation**. Le cône d'accessibilité s'élargit avec le temps (contour **rouge**, ouvert à droite) et permet d'atteindre de plus en plus de fonctionnalités. Un autre cône, tourné vers le passé (contour **vert**, ouvert à gauche), modélise ce qui peut être fait avant la fin du *sprint* ( $t_i$ ). On voit des fonctionnalités qui valent la peine d'être commencée (point **vert**) car il est possible de les terminer avant la fin du *sprint*, et d'autres (point **rouge**) qu'on pourrait commencer, mais pas terminer.



Les préparatifs de la démo qui termine souvent chaque *sprint* doivent donc être inscrits dans ce diagramme, que ce soit en **analyse**, **développement** ou **intégration et validation**. Par ailleurs, d'autres démos peuvent avoir lieu tout le long du *sprint* pour s'expliquer, se faire comprendre, ou convaincre. Ces démos imprromptues seront d'autant plus faciles à faire que l'infrastructure de démo aura été intégrée au développement.