

Septembre Exclusif | Acheter 2, Obtenez la 3ème gratuite sur les cours en ligne en direct

[Inscrivez-vous aujourd'hui](#)



Divulgateion progressive



Jakob Nielsen

Le 3 décembre 2006

Nous apprécions votre vie privée



Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)

Résumé: La divulgation progressive reporte les fonctionnalités avancées ou rarement utilisées à un écran secondaire, ce qui rend les applications plus faciles à apprendre et moins sujettes aux erreurs.

Les concepteurs d'interaction font face à un dilemme:

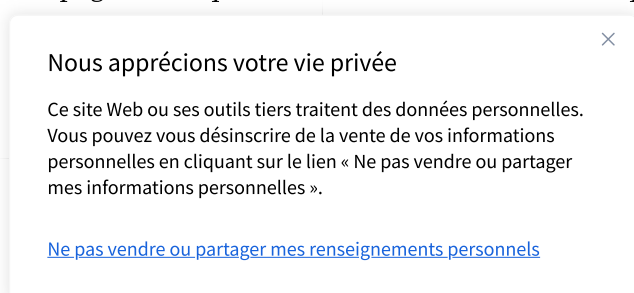
- Les utilisateurs veulent **de l'énergie**, des fonctionnalités et suffisamment d'options pour gérer tous leurs besoins spéciaux. (*Tout le monde* est un cas spécial en quelque sorte. Par exemple: Qui veut des numéros de ligne dans un traitement de texte? Des millions d'utilisateurs, c'est qui, y compris la plupart des grands cabinets d'avocats.)
- Les utilisateurs veulent de **la simplicité**; ils n'ont pas le temps d'apprendre une profusion de fonctionnalités en assez profondeur pour sélectionner les quelques-uns qui sont optimaux pour leurs besoins.

La divulgation progressive est l'un des meilleurs moyens de satisfaire à ces deux exigences contradictoires. C'est une idée simple, mais puissante:

1. Initialement, affichez aux utilisateurs **seulement quelques-unes** des options les plus importantes.
2. Offrez un **ensemble plus grand** d'options spécialisées sur demande. Ne divulguez ces fonctionnalités secondaires que si un utilisateur les demande, ce qui signifie que la plupart des utilisateurs peuvent poursuivre leurs tâches sans se soucier de cette complexité supplémentaire.

La **boîte de dialogue d'impression** est l'exemple classique de la divulgation progressive. Lorsque vous émettez la commande pour imprimer un document, vous obtiendrez une boîte de dialogue avec un petit ensemble de choix - principalement, combien de copies à imprimer, mais peut-être quelques autres variantes, telles que l'impression de l'ensemble du document ou un sous-ensemble, et quelle imprimante utiliser. Malheureusement, les boîtes de dialogue d'impression ont augmenté au cours de la dernière décennie, et certaines applications offrent une boîte de dialogue initiale avec des options très détaillées qui seraient mieux placées dans une boîte de dialogue secondaire.

La boîte de dialogue d'impression initiale contient généralement un ou plusieurs boutons pour les options avancées. Ces boutons conduisent à des dialogues secondaires qui permettent aux utilisateurs de spécifier des paramètres rarement utilisés, tels que la mise à l'échelle et l'impression des pages en séquence inverse. Si l'utilisateur clique sur le bouton d'options avancées, le système



Applications vs Sites Web

Applications vs. Sites Web

La divulgation progressive a longtemps été l'une des [principales lignes directrices de la conception d'applications](#), car la plupart des applications ont tellement de commandes, de fonctionnalités et d'options qu'il est logique d'en différer vers une zone secondaire. Les sites Web sont devenus si complexes que [la divulgation progressive est une bonne idée pour de nombreux sites riches en information](#). [Différer le matériel secondaire est également une ligne directrice clé pour la conception mobile](#).

Un site de commerce électronique, par exemple, peut mentionner quelques attributs clés du produit sur la [page](#) principale du [produit](#) et laisser les utilisateurs cliquer sur une [page](#) secondaire pour voir les spécifications complètes du produit.

Hypertext fournit une implémentation simple de la divulgation progressive: les pages de niveau supérieur contiennent des concepts de niveau supérieur et des descriptions simplifiées, et les pages de niveau inférieur remplissent les détails pour les utilisateurs qui veulent tout savoir.

Avantages de la divulgation progressive

Dans un système conçu avec une divulgation progressive, le fait même que quelque chose apparaisse sur l'affichage initial indique aux utilisateurs qu'il est **important**.

Pour les **utilisateurs novices**, cela aide à prioriser leur attention afin qu'ils ne passent du temps que sur des fonctionnalités les plus susceptibles de leur être utiles. En cachant les paramètres avancés, la divulgation progressive aide les utilisateurs novices à éviter les erreurs et leur évite le temps qu'ils auraient passé à envisager des fonctionnalités dont ils n'ont pas besoin.

Pour les **utilisateurs avancés**, l'affichage initial plus petit leur fait également gagner du temps car ils évitent d'avoir à scanner une grande liste de fonctionnalités qu'ils utilisent rarement.

La divulgation progressive améliore ainsi 3 des [5 composants](#) de l'utilisabilité: l'apprentissage, l'efficacité d'utilisation et le taux d'erreur.

Vous pouvez supposer qu'en concentrant initialement l'attention des utilisateurs sur quelques fonctionnalités de base, ils pourraient construire un modèle mental limitant du système et donc être incapables de comprendre toutes leurs options. [La recherche dit qu'il s'agit d'inquiétudes sans fondement](#): les gens comprennent *mieux* un système lorsque vous les aidez à prioriser les

Nous apprécions votre vie privée

Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)

plus importantes.

Divulcation progressive

deux choses que **vous devez bien comprendre** lors

de la conception pour la divulgation progressive:

- Vous devez obtenir la bonne **division entre les caractéristiques initiales et secondaires**. Vous devez divulguer tout ce dont les utilisateurs ont souvent besoin à l'avance, de sorte qu'ils doivent passer à l'affichage secondaire seulement à de rares occasions. À l'inverse, la liste principale ne peut pas contenir trop d'options ou vous ne concentrerez pas suffisamment l'attention des utilisateurs sur des questions vraiment importantes. Enfin, l'affichage initial ne peut pas contenir de fonctionnalités confuses ou vous ralentirez les performances de l'utilisateur.
- Ça doit être **évident comment les utilisateurs progressent** des niveaux de divulgation primaire à secondaire:
 - Tout d'abord, simplifier la **mécanique** de cette opération. Pour un site Web, suivez les directives pour [visualiser les liens](#). Pour une application, placez le bouton de fonctionnalités avancées dans un endroit clairement visible.
 - Deuxièmement, étiquetez le bouton ou le lien d'une manière qui définit **des attentes** claires pour ce que les utilisateurs trouveront lorsqu'ils passeront au niveau suivant. (En d'autres termes, la progression devrait avoir une forte [odeur d'information](#).)

En ce **qui** concerne le premier point, comment déterminez-vous **quelles caractéristiques le niveau de divulgation initial devrait contenir**? L'analyse des tâches et les études de terrain peuvent vous donner un aperçu de ce dont les gens ont besoin, même si vous travaillez sur un nouveau design. Si vous améliorez un système existant, les statistiques de fréquence d'utilisation peuvent vous aider à prioriser les fonctionnalités. Pour un site Web, [les analyses de base](#) vous indiquent à quelle fréquence les gens regardent diverses pages. Cependant, vous devez compléter de telles analyses avec des [tests d'utilisabilité observationnelle](#) pour discerner si une page reçoit de nombreux succès parce que les utilisateurs le *veulent* ou parce qu'ils entrent simplement dans la page par *erreur*. Pour une application, vous pouvez obtenir des données d'utilisation encore plus détaillées en instrumentant le code pour enregistrer la fréquence à laquelle les gens utilisent diverses fonctionnalités.

In theory, there's no reason why you can't have **multiple levels of progressive disclosure**. Even though the secondary level is for experienced users, there are still limits to how complex it should be; some options might be so specialized or rare that you should relegate them to a tertiary level. In practice, designs that go beyond 2 disclosure levels typically have low usability because users often get lost when moving between the levels.

If you have so many features that you need 3 or more levels, consider [simplifying your design](#). If you

Nous apprécions votre vie privée

Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)



...ir advanced features into groups that make sense, ignore any areas that they don't need. Use grouping right, and invest extra time in [user testing](#) performance, including both common tasks and

It's also possible to have **multiple secondary displays**, each of which is revealed by a different

control on the initial display. The obvious upside to this is that you can accommodate more features without introducing a tertiary level. The obvious downside is added complexity on the initial level, since users must consider multiple buttons for advanced options.

Because progressive disclosure's goal is to expedite use of the initial display, it's rarely a good idea to offer multiple ways to progress to secondary options. If you must do this, however, at least ensure that you clearly label all choices.

Staged Disclosure: One Step at a Time

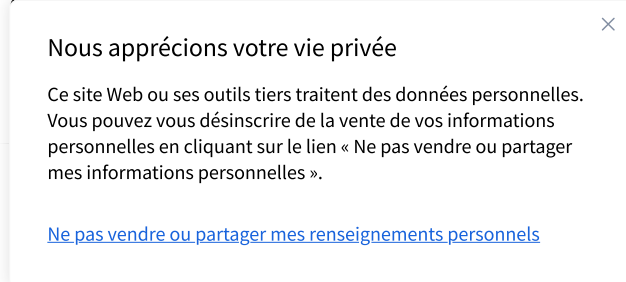
Staged disclosure is a variant in which users step through a linear sequence of options, with a subset displayed at each step. **Wizards** are the classic example of staged disclosure. The differences between progressive disclosure and staged disclosure are:

	Progressive Disclosure	Staged Disclosure
Initial display	Core features	Features that users access first in the task sequence
Subsequent display(s)	Secondary features	Features that users access later in the task, even if these features are equally important (or more important)
Do users access subsequent displays?	Usually not — most users get what they need on the initial display	Yes — unless users stop the task before completing the sequence
Navigation between displays	Hierarchical: users start at the initial display and, if necessary, move to the secondary display and then (often) return to the initial display	Linear: users progress through the task one step at a time
Main usability benefit	Learnability: novice users are focused on the most useful features and confusingly advanced features are hidden	Simplicity: each step is simple and its purpose is clear because features that belong to other steps are hidden

Staged disclosure is **useful** when you can divide a task into distinct steps that have little interaction. It is **problematic** when the steps are interdependent and users must alternate between them.

When researching the usability of [46 web-based applications](#), we tested a hotel reservation system that squeezed all the reservation stages onto a single screen. Comparing this design with most hotel sites — which distribute the reservation stages over multiple pages — shows the benefits and

drawbacks of staged disclosure:



users were trying to decide which hotel room to get availability and prices for different hotel room offered by the user. Most hotel sites show room across multiple pages, making it more difficult to see all the options. However, because most people use all of these features together, with

frequent back-and-forth modifications, so they should be on 1 screen.

- The single-screen design also caused usability problems because it included a segment for users to enter their address and credit card information. Hotels need such information to complete a booking, but it's not needed during the exploratory phase when users are experimenting with various reservations. Staged disclosure would defer the payment details to a secondary screen. Doing so would leave more space on the first screen for room information and the exploratory interface, thus protecting users from various errors that result when too many options are displayed at the same time.

As this example shows, staged disclosure requires a thorough task analysis to understand which options are used together and which are better thought of as separate stages. The more features you can defer, the simpler your design, but if you divide the task into too many steps, users get bogged down by excess navigation.

The single-screen hotel reservation design would have been better as a 2-screen design — something overlooked by many naïve analyses that consider 1 screen vs. 5 screens to be the only choices. Testing the 1-screen design with real users (which the hotels don't seem to have done) quickly shows that the compromise of a 2-screen solution would be better.

Managing Complexity

Progressive disclosure and staged disclosure are both strategies to manage the [profusion of features](#) and options in modern user interfaces. They are both more than 30 years old and have proven themselves useful in countless applications, and even some websites. Try them — but stay aware of their weaknesses.

Cours connexes

Conception d'application pour Web et Desktop

Composants, modèles de conception, flux de travail et façons d'interagir avec des données complexes

▲ INTERACTION

Nous apprécions votre vie privée

Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)

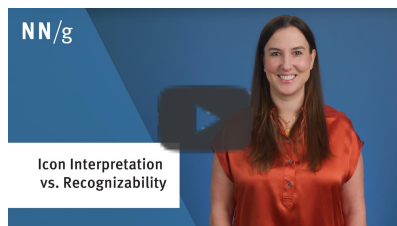
NNgroup



Regarder sur

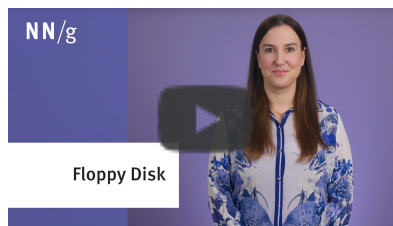
Progressive Disclosure

Raluca Budiu · 5 min



Icon Interpretation vs. Recognizability

Kate Kaplan · 4 min



Is the Floppy Disk Dead?

Kate Kaplan · 4 min



2-Factor Authentication (2-FA)

Tim Neussesser · 4 min

Related Articles:

Top 10 Application-Design Mistakes

Jakob Nielsen and Page Laubheimer · 13 min

6 Tips for Successful Personalization

Amy Schade · 6 min

Wireflows: A UX Deliverable for Workflows and Apps

Page Laubheimer · 6 min

Nous apprécions votre vie privée

Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)

✕ s. Radio

6 min

Indicators, Validations, and Notifications: Pick the Correct Communication Option

Kim Flaherty · 9 min

Never miss an update

Get weekly UX articles, videos, upcoming courses, and job openings straight to your inbox with our newsletter.

[Subscribe](#)

Courses

[Live Online Courses](#)[Self-Paced Courses](#)

Consulting

[Intensive Applied Workshops](#)[Customized Research](#)[Expert Review](#)[Keynote Speaking](#)[User Testing](#)[UX Maturity Assessment](#)

Help

[Terms & Conditions](#)[Privacy Policy](#)[Live Course Tech Support](#)[FAQs](#)

Company

[About NNGroup](#)[People](#)[News & Careers](#)[Contact Us](#)

Copyright © 1998-2026 Nielsen Norman Group, All Rights Reserved.

[Cookie Preferences](#) | [Cookie Declaration](#)

Follow Us

[Youtube](#)[LinkedIn](#)[Instagram](#)[Threads](#)[Bluesky](#)[X](#)[Facebook](#)[Spotify](#)

Nous apprécions votre vie privée

Ce site Web ou ses outils tiers traitent des données personnelles. Vous pouvez vous désinscrire de la vente de vos informations personnelles en cliquant sur le lien « Ne pas vendre ou partager mes informations personnelles ».

[Ne pas vendre ou partager mes renseignements personnels](#)