

# RESEET

LE SERIOUS GAME DE L'ECO-CONCEPTION DE SERVICES NUMERIQUES

Kit participant

Capgemini  invent

1. Rappel des objectifs pédagogiques de Reset
2. La principale ressource de Reset : le GR491
3. Les règles d'écoconception à retenir, par métier
4. Que faire en aval de la session

# RÉSET

LE SERIOUS GAME DE L'ECO-CONCEPTION DE SERVICES NUMERIQUES

## Rappel des objectifs pédagogiques de Reset



### Un jeu d'apprentissage

Chaque représentant métier doit **s'approprier les bonnes pratiques d'écoconception liées à son domaine d'activité**, et comprendre les rôles et interactions nécessaires de tous les métiers représentés pour écoconcevoir un service numérique.

-> Elles sont listées dans les pages suivantes.

Pour rappel, l'**écoconception**, c'est **intégrer les impacts environnementaux et sociaux du numérique dans la conception d'un service numérique**, et viser directement ou indirectement à allonger la durée des vies des équipements, à réduire la consommation de ressources informatiques et énergétiques des terminaux, des réseaux et des centres de données.

## Rappel des objectifs pédagogiques de Reset



### Les éléments mis en place pour sécuriser l'apprentissage malgré l'enjeu de vitesse

Le passage d'une salle de réparation à une autre est possible seulement si **toutes les réponses sont bonnes**. Sinon, l'équipe doit « payer » la réparation contre une pénalité de temps...

Des réponses pièges ont été glissées pour inciter les participants à lire l'ensemble des éléments figurant dans le plateau de jeu

Un maître du jeu (le joueur connaissant le mieux le thème de la salle de réparation) doit lire à voix haute l'ensemble des éléments figurant dans la salle

## La principale ressource de Reset : le GR491

# 1

### Qu'est-ce que le GR491 ?

Le GR491 est le guide de référence de conception responsable de services numériques proposé par l'Institut du Numérique Responsable – ce dernier estimant qu'une conception responsable des services numériques est un levier majeur d'un numérique plus durable, accessible et respectueux de la vie privée.

Il s'adresse à toutes organisations, de toute taille, qui conçoivent et développent des services numériques, soucieuses d'intégrer des critères liés à l'éco-conception et à l'accessibilité dans les projets qu'elles mènent. Il est basé sur le triptyque :

- People : Réduire l'empreinte environnementale du numérique
- Planet : Réduire la fracture sociale et améliorer les conditions de vie grâce au numérique
- Prosperity : Le numérique en tant que levier de croissance, doit être une opportunité pour toutes et tous

Il contient 491 critères divisés en 8 familles et s'adresse à l'intégralité des équipes d'un projet, que ce soit la direction, les achats, les RH ou les métiers de la DSI.

Il est accessible ici <https://gr491.isit-europe.org/>

## La principale ressource de Reset : le GR491

# 2

### Comment le GR491 a servi pour construire ce Serious Game ?

Le serious game a été construit **sur la base du GR491** et adapté afin d'offrir un support pédagogique optimal.

Aussi, les 491 critères ont été **retravaillés** - priorisés, mutualisés, reformulés - à l'aide du comité scientifique et du réseau d'experts Capgemini et freelance.

**La priorisation a été réalisée en :**

- Retenant systématiquement les critères identifiés comme « **incontournables** » dans le référentiel
- Dégageant les **critères les plus impactants** d'un point de vue Planet / People / Prosperity (score le plus élevé)

A noter que la dimension Planet est la plus mise en avant car considérée comme prioritaire pour la majorité des organisations.

1. Inclure la démarche Numérique Responsable dès l'origine du projet.
2. Exprimer les fonctionnalités en terme de besoins vs envies et les confronter aux réalités terrain dans les ateliers de cadrage.
3. Identifier le parcours usager/processus de bout en bout et faire ressortir les recoupements, doublons de fonctionnalités et incohérence dans leur relation
4. Installer un processus de revue Numérique Responsable dans la méthodologie projet avec une démarche d'amélioration continue. Permettre que le point de vue de chacun soit intégré, tout en les sensibilisant au Numérique Responsable. Ne pas hésiter à faire intégrer tous les acteurs à la démarche agile et/ou Design Thinking ou toute démarche collaborative.
5. Evaluer les bénéfices extra-financiers des décisions prises tout au long de la réalisation. Se servir des indicateurs RSE et de performance NR définis en début de projet afin d'en assurer le suivi tout au long du cycle de vie du projet. -> Evaluer l'économie en temps gagné/non dépensé en développement (ROI) grâce aux bonnes pratiques du NR (reuse de composants / fonctionnalités, utilisation no code, low code...).

1. Valider que la finalité du projet répond au moins à l'un des 17 Objectifs de Développement Durable (ODD)
2. Inclure la démarche Numérique Responsable dès l'origine du projet
3. Communiquer les enjeux NR du projet à l'ensemble des parties prenantes internes et externes dans la phase de lancement
4. Exprimer les fonctionnalités en terme de besoins vs envies et les confronter aux réalités terrain dans les ateliers de cadrage
5. S'assurer que les équipes MOA et MOE disposent de la connaissance technique et NR pour mener à bien le projet sur l'ensemble du cycle de vie du service
6. Installer un processus de revue Numérique Responsable dans la méthodologie projet avec une démarche d'amélioration continue
7. Définir les enjeux de communication de la démarche NR vers l'ensemble des parties prenantes y compris les utilisateurs en fournissant un document de « déclaration de conformité NR » sur le modèle de la « déclaration d'accessibilité »
8. Anticiper la fin de vie des éléments matériels et logiciels générés et utilisés lors du projet avec des procédures associées aux mécanismes de suppression de chaque composant ou donnée
9. Définir et partager la liste des indicateurs RSE et de performance NR suivis tout au long du cycle de vie du projet
10. Evaluer les bénéfices extra-financiers des décisions prises tout au long de la réalisation

1. Valoriser les équipements en fin de vie & préférer des équipements réparables ou évolutifs lors de l'achat
2. Préférer un datacenter au PUE (Power Usage Effectiveness) le plus faible possible. La cible est entre 1,2 et 1,4
3. Choisir la localisation de son data center ou le lieu de collecte des données (edge) en fonction de la localisation de ses utilisateurs
4. Rendre disponibles les caractéristiques (Ancienneté, données de performance environnementales, ...) pour chaque équipement
5. S'assurer que le dimensionnement des ressources d'infrastructures suit la vie de l'application

1. Valoriser les équipements en fin de vie & préférer des équipements réparables ou évolutifs lors de l'achat
2. Assurer un suivi des ressources utilisées (notamment la consommation d'eau) des data centers
3. Utiliser des serveurs avec des normes, labels et certifications environnementales reconnues
4. Valoriser ou recycler les déchets issus des data centers et vérifier la filière de traitement des déchets
5. Préférer un datacenter au PUE (Power Usage Effectiveness) le plus faible possible. La cible est entre 1,2 et 1,4
6. S'assurer que le système de sauvegarde est incrémentiel et identifier le volume des sauvegardes quotidiennes
7. Choisir la localisation de son data center ou le lieu de collecte des données (edge) en fonction de la localisation de ses utilisateurs
8. S'assurer que l'ensemble des équipements techniques utilisés par le service ont été identifiés
9. Rendre disponibles les caractéristiques (Ancienneté, données de performance environnementales, ...) pour chaque équipement
10. S'assurer que le dimensionnement des ressources d'infrastructures suit la vie de l'application

1. Mettre en place les éléments permettant le contrôle de la qualité NR des logiciels produits ainsi que les outils pour suivre l'utilisation effective lors de la production.
2. Choisir les solutions techniques en fonction de la capacité des fournisseurs et/ou intégrateurs à exposer leurs politiques RSE et le respect des enjeux NR
3. Limiter les besoins en stockage et en flux de données : optimiser la redondance, limiter la volumétrie et sa croissance, utiliser des technologies de cache et de CDN. Utiliser un format (modèle) de données pertinent (stockage et transmission). Ex. éviter le décodage entre le système existant et le nouveau système. ex. Favoriser les modèles efficaces (YAML, JSON, binaire...) vs inefficaces (XML...)
4. Adapter les ressources au plus juste à la demande tout en respectant les exigences métier : Adapter les besoins de ressources de traitement instantanées (« temps réel ») au besoin utilisateur, privilégier les mécanismes permettant de diluer dans le temps les charges opérationnelles. Optimiser les ressources (serveur & terminal) nécessaires pour l'application avec le respect des niveaux de service (SLA) attendus et adaptés pour chaque composant
5. Penser chaque composant logiciel et matériel pour être réutilisable, autonome, générique, modulaire, en suivant les normes du marché

1. Lister les recommandations NR associés à appliquer sur l'ensemble des phases de vie du projet (conception, réalisation, déploiement, exploitation, décommissionnement), anticiper les indicateurs et moyens de mesure, prévoir des évaluations régulières
2. Mettre en place les outils de collecte et suivis des indicateurs NR et RSE dans les chaînes de DevOps et en production
3. Mettre en place les éléments permettant le contrôle de la qualité NR des logiciels produits ainsi que les outils pour suivre l'utilisation effective lors de la production
4. Choisir les solutions techniques en fonction de la capacité des fournisseurs et/ou intégrateurs à exposer leurs politiques RSE et le respect des enjeux NR
5. Définir les gammes d'équipements permettant de prolonger les durées d'utilisations des matériels et couvrant l'intégralité de la durée de vie du service produit par le projet
6. Optimiser les ressources (serveur & terminal) nécessaires pour l'application avec le respect des niveaux de service (SLA) attendus et adaptés pour chaque composant
7. Construire, stocker, présenter ou publier des données de manière à permettre que les données soient faciles à trouver, accessibles, interopérables et réutilisables (principe FAIR : Findable, Accessible, Interoperable, Reusable)
8. Limiter les besoins en stockage et en flux de données : optimiser la redondance, limiter la volumétrie et sa croissance, utiliser des technologies de cache et de CDN.
9. Adapter les besoins de ressources de traitement instantanées (« temps réel ») au besoin utilisateur, privilégier les mécanismes permettant de diluer dans le temps les charges opérationnelles
10. Penser chaque composant pour être réutilisable, autonome, générique, modulaire, en suivant les normes du marché

1. Mettre en place des outils de mesure de la consommation énergétique des algorithmes
2. Minimiser le temps de fonctionnement des clusters et éteindre les nœuds dans le cas d'une non-activité
3. Utiliser un langage compilé afin de minimiser la complexité algorithmique du code développé et structurer ses codes d'analyse et de traitement de données en petites fonctions pour un tout éventuel re-use des données
4. Se répertorier à la littérature scientifique afin de trouver des retours d'expériences semblables à l'expérimentation à faire avant de procéder aux tests de ces modèles sur son propre jeu de données
5. Eviter d'avoir des duplicata de données dans la procédure de téléchargement ou de stockage, sauvegarder en Cloud les données qui ne sont plus traités en format compressé (et par la suite, les mettre à jour), et favoriser le traitement on-fly en amont du téléchargement des données quand c'est possible

1. Mettre en place des outils de mesure de la consommation énergétique des algorithmes
2. Mettre à jour les librairies de programmation et le code qui lui est associé tout en supprimant les installations d'anciennes versions de librairies
3. Minimiser le temps de fonctionnement des clusters et éteindre les nœuds dans le cas d'une non-activité
4. Utiliser un langage compilé afin de minimiser la complexité algorithmique du code développé
5. Eviter d'avoir des duplicata de données dans la procédure de téléchargement ou de stockage et favoriser le traitement on-fly en amont du téléchargement des données quand c'est possible
6. Se répertorier à la littérature scientifique afin de trouver des retours d'expériences semblables à l'expérimentation à faire avant de procéder aux tests de ces modèles sur son propre jeu de données
7. Coder selon les normes du langage utilisé (minimiser les noms des variables recommandés) et documenter le code pour sa réutilisation et traiter les données en petites fonctions pour un tout éventuel re-use des données
8. Favoriser le traitement on-fly en amont du téléchargement des données quand c'est possible
9. Structurer ses codes d'analyse et de traitement de données en petites fonctions pour un tout éventuel re-use des données
10. Sauvegarder en Cloud les données qui ne sont plus traités en format compressé (et par la suite, les mettre à jour)

1. Effectuer une analyse du profil de consommation de l'application
2. Minimiser la complexité algorithmique du code développé
3. Ne pas allouer de ressources inutilement et libérer les ressources inutilisées dès que possible
4. Documenter le code source produit (commentaires, Doc As Code) pour favoriser sa pérennité et sa réutilisabilité
5. Minimiser les dépendances aux logiciels tiers ou veiller à ce que ces derniers soient « vertueux » (peu consommateurs de ressources, pérennes et peu intrusifs)

1. Effectuer une analyse du profil de consommation de l'application
2. Minimiser la complexité algorithmique du code développé
3. Ne pas allouer de ressources inutilement et libérer les ressources inutilisées dès que possible
4. Documenter le code source produit (commentaires, Doc As Code) pour favoriser sa pérennité et sa réutilisabilité
5. Minimiser les dépendances aux logiciels tiers ou veiller à ce que ces derniers soient « vertueux » (peu consommateurs de ressources, pérennes et peu intrusifs)
6. Veiller à la modularité du code produit pour favoriser son évolutivité et sa réutilisabilité
7. Limitier les données au strict nécessaire à chaque stade de leur traitement (calcul, échange, stockage)
8. Envisager une compression des données lorsque l'empreinte mémoire, disque ou réseau est plus importante que le temps processeur
9. Veiller à l'optimisation des ressources entre le client et le serveur
10. Concevoir l'application en « Mobile first »

1. Se concentrer sur la conception de l'unité fonctionnelle principale du service numérique, retirer toutes les fonctionnalités superflues.
2. Optimiser le parcours utilisateur afin qu'il soit le plus efficace que possible.
3. S'assurer que le service fonctionne avec des connexions faibles et des appareils "anciens".
4. Compresser, optimiser tous les médias et documents. Pas de lecture automatique pour les médias. Animations, vidéos et sons ne sont pas en lecture automatique (auto-play) mais lancés/activables par l'utilisateur.
5. Privilégier les contenus média à caractère informatif plutôt que décoratif. Informer l'utilisateur de l'impact environnemental de la conception ou des fonctionnalités proposées.

1. Se concentrer sur la conception de l'unité fonctionnelle principale du service numérique, retirer toutes les fonctionnalités superflues
2. Optimiser le parcours utilisateur afin qu'il soit le plus efficace que possible
3. S'assurer que le service fonctionne avec des connexions faibles et des appareils "anciens"
4. Compresser, optimiser tous les médias et documents
5. Privilégier les contenus média à caractère informatif plutôt que décoratif
6. Animations, vidéos et sons ne sont pas en lecture automatique (auto-play,) mais lancés/activables par l'utilisateur
7. Éviter, ou réduire au maximum, l'utilisation de services tiers
8. Ecrire des specs stratégiques, mais présenter aux utilisateurs l'impact environnemental du service ou des fonctionnalités
9. Laisser la main aux utilisateurs sur les notifications qu'ils reçoivent
10. Limiter le nombre de polices de caractères (fonts) et les variantes de polices appelées (graisse, caractères utilisés dans le projet); privilégier les polices standard et limiter les polices propriétaires (ou bien limiter les variantes de ces polices)

Avez-vous participé à un session de jeu-sensibilisation RESET ?



1. Obtenez votre badge participant via [ce lien](#) !

2. Partagez-le sur les réseaux sociaux !



3. Donnez votre avis via ce [formulaire de satisfaction](#) !

Vous avez une question ou souhaitez aller plus loin ?

# RÉSET

LE SERIOUS GAME DE L'ECO-CONCEPTION DE SERVICES NUMERIQUES

Contactez [reset@capgemini.com](mailto:reset@capgemini.com) !