

# **Le cahier des charges technique : détailler la mise en œuvre**

# Table des matières

<b>I. Décrire la solution sous l'angle technique</b>	<b>3</b>
A. Présenter et justifier la solution technologique .....	3
1. Grands axes technologiques .....	3
2. Identification de la solution .....	5
3. Présenter la couverture fonctionnelle .....	6
B. Présenter l'architecture technique attendue.....	7
C. Compléter les spécifications fonctionnelles .....	8
<b>II. Exercice : Quiz</b>	<b>9</b>
<b>III. Préciser les spécificités de la réalisation</b>	<b>10</b>
A. Présenter l'environnement de développement.....	10
1. Environnement de développement, pour produire.....	10
2. Environnement de préproduction, pour tester .....	11
3. Environnement de production, pour livrer.....	11
4. Détail de l'environnement de développement.....	11
B. Lister les contraintes et procédures techniques.....	12
1. Contraintes de structuration et bonnes pratiques.....	12
2. Modalités de livraison et d'archivage.....	14
C. Présenter l'hébergement .....	15
D. Anticiper la maintenance et les évolutions.....	16
<b>IV. Exercice : Quiz</b>	<b>17</b>
<b>V. Essentiel</b>	<b>19</b>
<b>VI. Auto-évaluation</b>	<b>19</b>
<b>Solutions des exercices</b>	<b>21</b>

## I. Décrire la solution sous l'angle technique

### Contexte

Votre **cahier des charges fonctionnel** décrit **dans les détails**, et dans un **contexte éclairé**, la **solution** qui sera **réalisée en cours de projet**. Il constitue un **référentiel** de ce qui devra être développé, mais aussi un **guide pour tous les acteurs** du projet.

Un **projet digital** revêt **nécessairement** une **dimension technologique** qui devra également être adressée. Vous pourrez le faire grâce au **cahier des charges technique**. Tout comme le cahier fonctionnel, ce sera un **document de référence** pour votre projet.

**Orienté pour les développeurs**, il fera la **synthèse des choix technologiques** adoptés, présentera les **contraintes** et les **modes de développements** retenus.

Le cahier des charges technique peut être **intégré à votre cahier des charges fonctionnel** pour constituer une **somme de référence** de votre projet. Destiné à un **public plus ciblé**, il peut également **faire l'objet d'un document séparé**.

Dans tous les cas, il doit partager des **informations spécifiques** dont nous allons parcourir **les grandes lignes** dans ce cours.

Le **cahier des charges fonctionnel** décrit **l'organisation** de votre solution, les **contenus** qu'elle devra proposer et comment **les utilisateurs** pourront **interagir** avec elle. Le **cahier des charges technique** s'attachera également à **documenter la solution attendue**, mais sous un angle **strictement technique**.

Les **deux documents** sont **complémentaires** et sont d'ailleurs souvent **regroupés en un seul**. Les **développeurs** auront besoin de **connaître les attendus en termes d'usages** pour **bien identifier** ce qu'ils devront réaliser. Mais les **créatifs**, par exemple, et les **concepteurs** en général, devront **connaître les solutions technologiques retenues** dans le cadre du projet : elles pourront parfois **influencer** ou **contraindre** les **choix créatifs ou fonctionnels**.

### A. Présenter et justifier la solution technologique

Dans le **cahier des charges fonctionnel**, vous aurez pu indiquer que **le site** imaginé **reposera sur un CMS** afin d'en **faciliter la gestion des contenus**. Le **cahier des charges technique** viendra **préciser** (et **justifier**) **lequel**.

Vous aurez peut-être aussi indiqué, **dans vos contraintes**, que votre réalisation devra être **accessible** et **respecter les standards du Web** ; le **cahier des charges technique** indiquera **précisément** les **langages** à utiliser.

Il aura donc vocation à **compléter votre description du « quoi »**, mais en fournissant des **informations utiles** à vos **équipes de développement**, mais aussi aux **équipes techniques** de votre **commanditaire**.

#### 1. Grands axes technologiques

Une des **grandes questions** qui se pose pour un **développement Web**, c'est le **choix de la plateforme** déployée et des **langages** sur lesquels elle repose. Vous allez ainsi devoir **expliquer** si vous allez recourir à un **CMS** (*Content Management System*) ou réaliser un **développement sur-mesure**. Et parfois, pourquoi pas, **les deux en même temps**.

### Exemple

Pour **présenter votre approche de développement**, vous allez devoir **détailler** quelles sont les **raisons** qui vous ont amenés à faire ce **choix**. Par exemple :

#### Envisager des développements sur-mesure

- Éviter le **code superflu** conçu pour des **fonctionnalités inutilisées** dans le cadre du projet,
- Garder **la maîtrise** sur le code développé,
- Construire une plateforme **moins vulnérable aux attaques**, car différente des celles largement distribuées,
- Pouvoir implémenter des **fonctionnalités originales**,
- Etc.

#### Utiliser une plateforme CMS existante

- Bénéficier d'une **solution déjà faite** pour **réduire significativement les temps de développement**.
- Pouvoir s'appuyer sur une **communauté d'entraide** pour résoudre d'éventuels problèmes.
- Bénéficier d'une **maintenance existante** pour réagir vite en cas de **faille de sécurité**.
- S'appuyer sur une **solution connue** qui pourra être **maintenue facilement** par une **équipe différente**.
- Etc.

#### S'appuyer sur un CMS complété par des développements spécifiques.

En plus des **atouts liés aux deux options précédentes**, vous pourriez dans ce cas ajouter que cette **démarche mixte** sera utile pour :

- S'assurer d'un **socle éprouvé** pour les **fonctionnalités usuelles**,
- Autoriser des **développements sur-mesure** pour des **fonctionnalités inédites** ou très **spécifiques**,
- Permettre des **interactions poussées** avec d'autres éléments du **système d'information**,
- Améliorer **l'interface d'administration** pour la rendre **utilisable** par des profils néophytes,
- Etc.

Les **spécificités de votre projet** pourront bien entendu **alimenter votre argumentaire**, bien au-delà de ces **éléments génériques**. Gardez aussi à l'esprit que **chaque argument** va impliquer des **contreparties**, il faudra donc que **vos justifications soient solides**.

Cette question tranchée, vous devrez également devoir **préciser vos choix** concernant les **langages « socles »** que vous allez utiliser : **PHP** ? **.Net** ? **Python** ? Généralement, ces choix seront **très orientés par les expertises dont vous disposez** dans vos équipes, mais vous devrez également **les préciser** dans le cahier des charges technique.

Lorsque vous envisagez des **développements sur-mesure**, vous devrez aussi **faire un choix** dans **la manière de produire votre code**. Vos développeurs pourront le produire **de bout-en-bout** et concevoir chaque instruction, ou **s'appuyer sur des bibliothèques ou frameworks** déjà disponibles. Ces solutions sont **nombreuses** et permettent de **gagner un temps non négligeable** dans les développements.

### Exemple

Les **frameworks** et **bibliothèques** proposent des **fonctionnalités déjà développées** et **réutilisables** dans vos propres développements. C'est en cela qu'ils permettent de **produire plus vite**, même si le **fonctionnement** de chaque **framework** est **spécifique** et demandera une **phase d'apprentissage** aux développeurs qui ne le connaissent pas.

Parmi les **frameworks** répandus, vous pourrez rencontrer **Symfony** (*framework* PHP), **RoR** (*Ruby on Rails* écrit en Ruby), **JEE** (un *framework* Java) ou encore **NodeJS** (en JavaScript).

La question se posera de la même façon pour un **développement mobile** : allez-vous déployer des solutions **spécifiques pour chaque environnement** technologique (généralement *Apple* et *Android*, on parlera alors de **développement « natif »**) ou au contraire privilégier un développement dit « **hybride** » dans lequel vos développeurs travailleront sur un **code originel commun** reposant sur des **technologies Web** ?

Dans votre **cahier des charges technique**, vous devrez **présenter et argumenter vos choix directeurs**, comme vous le feriez dans le cahier des charges fonctionnel lorsque vous présentez vos **grands axes créatifs**. Il n'y aura pas, intrinsèquement, de bons ou mauvais choix, mais ils devront être **justifiés**.

#### Attention

Disons-le clairement : **il n'est nullement question que le chef de projet s'improviser lead développeur** et détermine **seul** les **choix technologiques**.

Sauf **contrainte explicite** de la part du **commanditaire**, ces choix reviennent aux « *professionnels de la profession* » : les **équipes de développement**. En revanche, le chef de projet veillera à ce que ces choix soient **effectivement faits** et **recueillera les arguments** nécessaires pour les **justifier** auprès des acteurs concernés.

Il pourra être chargé de la **réalisation** et du **formalisme** du **cahier des charges technique**, mais il est peu probable qu'il en définisse seul les **éléments de fond**. Dans certaines organisations, il existe le poste de **chef de projet technique**. Généralement **développeur de formation**, ce profil sera, lui, **légitime** pour **rédiger un cahier des charges technique de bout en bout**.

Cette précision reste valable **pour l'ensemble de ce cours**.

## 2. Identification de la solution

Si vous avez opté pour une **solution socle existante**, vous allez devoir **la présenter**. Dans le cadre d'un projet Web, il s'agira de dire **quel CMS aura été choisi**. Le **nombre de CMS** disponibles est **gigantesque**. Il existe des **solutions propriétaires** (payantes), mais également énormément de **solutions open source** distribuées gratuitement. Chacune se positionne, au moins historiquement, sur un **type de site** bien déterminé.

#### Exemple

Les CMS, *Wordpress* et *Dotclear* par exemple, sont historiquement **des moteurs de blog**. Ils gèrent donc particulièrement bien les sites avec **beaucoup de contenus** ou des **publications régulières**.

*Spip* est largement utilisé pour diffuser des **revues**, en particulier universitaires, car il est conçu pour **appliquer nativement les règles propres à l'édition** et favorise les **contributions** des utilisateurs.

*Magento* sera, lui, particulièrement adapté pour développer une **solution e-commerce**.

Dans les faits, chaque CMS permettant une **grande souplesse de personnalisation**, vous pourrez utiliser **à peu près n'importe lequel** pour développer votre projet. Tout en sachant que la **vocation première** de chaque CMS **influence fortement son utilisation et son paramétrage**.

Les options étant nombreuses, vous devrez **impérativement indiquer laquelle vous avez sélectionnée** et sur quels **critères**. Dans le cahier des charges technique, même si l'**information clé** est bien l'**indication de la solution choisie**, il vous faudra une nouvelle fois **justifier votre choix**. Une **comparaison chiffrée** pourra alors être très utile.

### Exemple

Imaginez que vous souhaitiez développer un **site Web e-Commerce**. Vous avez **opté pour une solution CMS**, mais vous avez le **choix entre plusieurs plateformes socles**. Vous allez alors **déterminer des critères pertinents** dans le cadre de votre projet et **évaluer chaque critère**, par exemple sur **une échelle de 1 à 5**. Vous pourriez ainsi obtenir un **tableau comparatif** ressemblant à celui-ci :

Critères	MAGENTO	DRUPAL COMMERCE	PRESTASHOP
<b>Complexité de paramétrage</b>	Complexe. Fonctionnement natif difficilement modifiable.	Moyen. Options de paramétrages nombreuses.	Simple et adaptable.
<b>Fiabilité de l'architecture</b>	Solide et éprouvée.	Solide et éprouvée.	Ne convient pas pour des projets d'envergure.
<b>Intégration au SI</b>	Difficile et peut performante, système d'API lent et complexe.	Très efficace, conçu nativement pour une intégration à des systèmes complexes.	API simple et flexible mais intégration limitée dans les SI lourds
<b>Disponibilité des ressources</b>	Peu de ressources disponibles, courbe d'apprentissage lente.	Ressources nombreuses (Drupal 7). Montée en compétence moyenne.	Ressources très nombreuses sur le marché, appropriation facile.
<b>Ergonomie du back office</b>	Moyenne.	Complexe.	Bonne.
<b>Gestion de contenus éditoriaux</b>	Fonctionnalités natives basiques.	Efficace, basée sur l'architecture Drupal.	Fonctionnalités natives basiques.
<b>Performances</b>	Moyennes.	Moyennes. Mais des composants peuvent alléger la charge serveur.	Limitées si nombre important de produits ou fort trafic.
<b>SCORE</b>	<b>16</b>	<b>23</b>	<b>19</b>

Ce formalisme rend **limpides** vos **critères de décision** et vous permet de **justifier** vos choix techniques.

### 3. Présenter la couverture fonctionnelle

Un de vos **éléments clés** pour **guider vos choix** sera la **couverture fonctionnelle**. Il s'agit là d'indiquer clairement **quelles sont les fonctionnalités** qui peuvent être **portées nativement** par la solution retenue et quelles seront celles qui **nécessitent des développements complémentaires**.

### Attention

**Plus la couverture fonctionnelle sera importante** pour la solution technologique que vous aurez retenue, plus elle apparaîtra comme **adaptée à votre projet**.

Pour vous aider à **matérialiser** cela, vous pourrez **vous appuyer sur votre périmètre fonctionnel** normalement déjà réalisé pour votre cahier des charges fonctionnel. Il vous suffira alors d'indiquer, **pour chaque fonctionnalité**, si elle peut être prise en charge **nativement** par votre socle technologique, si elle nécessite un **paramétrage** ou encore si vous devrez ajouter des **développements spécifiques**.

### Exemple

Cet exemple **reprend le périmètre fonctionnel** présenté dans le cahier des charges fonctionnel. Pour **chaque fonctionnalité** (et éventuellement **les gabarits de pages** lorsque c'est applicable), on a indiqué le **type de développement** attendu pour pouvoir concrétiser la fonctionnalité souhaitée.

Ref.	Type	Nom	Description	TYPE DE DÉVELOPPEMENT	MODULES COMPLÉMENTAIRES
D	Écran	Page produit	Page de présentation d'un produit	PARAMÉTRAGES	
D1	Fonctionnalité	Carrousel d'images	Présentation de 1 à 5 images par produit, avec vignettes et grande image	EXTENSION + PARAMÉTRAGES	CarrouselBuilder 3.1
D2	Fonctionnalité	Zoom image	Affichage de l'image en HD au clic sur l'image produit	DEV. SPÉCIFIQUE	
D3	Fonctionnalité	Téléchargement fiche produit	Téléchargement de la fiche produit au format PDF	NATIF	
X		Éléments récurrents	Éléments réutilisés		
X1	Zone répétée	Header	Présents sur toutes les pages du site	PARAMÉTRAGES	
X1.1	Fonctionnalité	Identifiant client	Présente le nom utilisateur ou un lien de connexion	NATIF	

Nous avons également **ajouté une colonne « modules complémentaires »** qui permet d'**identifier les extensions existantes** pour le CMS choisi. Notez que la plupart des grands CMS propose un **catalogue d'extensions**. Celles-ci peuvent parfois être **payantes** et vous devrez **en tenir compte dans l'établissement de votre budget**.

Cette représentation sera **particulièrement utile aux équipes** pour **identifier leurs tâches** et c'est également un outil précieux pour **produire des estimations de charge plus fiables**.

## B. Présenter l'architecture technique attendue

L'**architecture technique** que vous allez déployer est une **vision « macro »** de votre solution, mais **sous l'angle du développement**. Elle fera nécessairement **écho à votre solution fonctionnelle**, et notamment en s'appuyant sur **l'écosystème digital** que vous avez défini.

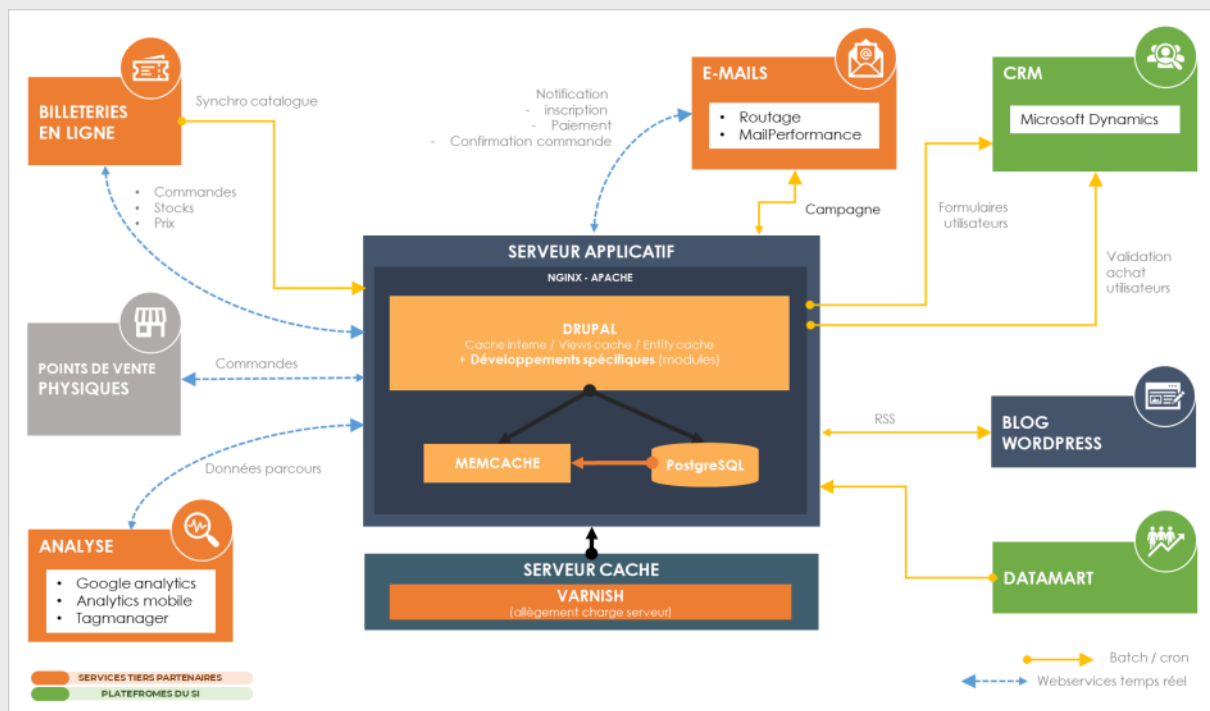
L'architecture technique doit **assurer le bon fonctionnement des échanges** entre les **différentes plateformes** réalisées dans le cadre du projet (vos **principaux chantiers**). Elle devra représenter, de façon **visuelle et synthétique**, tous les **liens** et les « **ponts** » entre vos plateformes.

Elle détaillera également **la nature des différentes plateformes** et des multiples **services et outils** mis en place dans le cadre du projet. Elle représentera les **types de serveurs** envisagés, les **systèmes de newsletters** avec lesquels la solution devra interagir, un éventuel **blog** distinct, les outils de **web-analyse** prévus, l'outil **CRM** déjà intégré au SI, etc. Sa composition **dépendra de votre projet**.

L'architecture technique **recense l'ensemble des choix** qui auront été effectués préalablement et **précise les interactions** attendues **entre l'ensemble des composants** de l'écosystème de manière à **assurer la cohérence et la stabilité** de l'ensemble.

### Exemple

Voici **un exemple d'architecture technique** qui pourrait être intégrée à votre cahier des charges :



Ce schéma présente **les serveurs Web** utilisés, le **CMS** choisi (**Drupal** ici) en mettant l'accent sur les **systèmes de cache** envisagés afin **d'optimiser les performances** du futur site. On indique que la **base de données** retenue sera **PostgreSQL**.

Le **blog** vient en complément du **site marchand** principal et fonctionnera sur une base **Wordpress**. Il sera **géré de façon autonome** même si **des liens sont prévus** entre les deux, notamment par l'intermédiaire de **flux RSS**.

Les **éléments représentés en vert** font partie du **système d'information** déjà existant du commanditaire. On peut par exemple remarquer **l'outil de gestion de la relation client** utilisé (**CRM**), **Microsoft Dynamics**. L'architecture indique ici que les **actions effectuées par les utilisateurs** sur le site Web **viendront alimenter le CRM**, notamment pour des **créations de compte** client ou lors de la **réalisation d'un achat**.

Les **éléments représentés en orange** sont des **services tiers** avec lesquels le site marchand principal devra pouvoir **interagir**. On représente ici des **services de billetterie** en ligne qui permettent de **vendre des places pour des événements**. On matérialise dans ce schéma que le **catalogue** sera **synchronisé dynamiquement**, tout comme les **stocks** (nombre de places disponibles) ou les **prix** de vente.

Si **l'écosystème digital** représente **la solution avec une approche stratégique et fonctionnelle**, le **schéma d'architecture technique** met en évidence les **plateformes technologiques** mobilisées et leurs **interactions** afin d'**assurer le fonctionnel** attendu.

Selon la **nature** de votre projet et le **niveau de détails** nécessaire, vous pourriez également dans cette partie expliquer **la structure de vos bases de données**, notamment en présentant un **schéma relationnel** par exemple.

### C. Compléter les spécifications fonctionnelles

Si votre cahier des charges fonctionnel comporte vos **spécifications fonctionnelles**, dans sa déclinaison **technique** vous devrez également préciser un certain nombre d'**informations indispensables aux développeurs**. Ce sera le cas par exemple **des vérifications attendues** sur des **données saisies par les utilisateurs** (vérifier **le format d'une adresse e-mail** par exemple) ou des **actions automatiques** qui peuvent se dérouler sans que l'utilisateur en ait connaissance.

#### Exemple

Considérons le cas d'un **site commercial** dans lequel vous prévoyez la **mise en place d'un jeu-concours** basé sur des « **instants gagnants** ».

Dans vos **spécifications fonctionnelles**, vous aurez simplement indiqué **comment sont conçus vos écrans** et ce que **l'utilisateur** pourra effectuer comme **action**. Par exemple, **côté fonctionnel** vous auriez pu préciser :

« **L'utilisateur active le levier et l'animation** bandit manchot est **lancée automatiquement**. En cas de **gain**, le **prix remporté** est révélé et il est **redirigé vers le formulaire d'inscription** pour saisir ses **coordonnées**. En cas d'**échec**, il est invité à **compléter le formulaire d'inscription** pour **jouer une seconde fois**. »

C'est **relativement clair**, mais **beaucoup de choses restent à préciser** pour être en mesure de **développer concrètement** cette fonctionnalité. Dans vos spécifications, vous devrez préciser des **informations essentielles pour les développeurs**. Par exemple :

«

- Les **instants gagnants**, générés **aléatoirement** et sous contrôle d'huissier avant le lancement de l'opération, sont **définis par une chaîne horodatée de type AAAAMMJJHHMMSSm.mm**. Chaque instant est **associé à une dotation, crypté et stocké** dans une table **dédiée et sécurisée**.
- **L'activation** de l'animation bandit manchot **déclenche un horodage** qui est **comparé à la base** des instants gagnants.
- La **comparaison** est **limitée à la seconde** pour définir si l'utilisateur est **vainqueur ou perdant**. En cas d'**activation simultanée** par plusieurs utilisateurs, le traitement est **géré en mode fille d'attente** et la **comparaison** est **portée à la milliseconde** de l'horodage. Le **premier** de la file remporte la dotation.
- **Toutes les participations sont enregistrées** dans des **logs séparés** et **stockés** de façon **sécurisée**. **L'IP utilisateur est enregistrée** ainsi que son **identification (ID\_user)** s'il est authentifié.



- Si l'utilisateur n'est **pas authentifié**, son **log de participation** sera **associé au compte créé à l'étape suivante**. Si ce processus est **abandonné**, le log recevra l'identifiant **ID\_Anonymous**.
- **Chaque nuit** pendant toute la durée de l'opération, une **tâche planifiée synchronise** le fichier des logs de participation.
- Etc.

»

Vous le constatez, **une action simple** du point de vue utilisateur peut **entraîner de multiples interactions et transferts de données**, importants à **spécifier dans le cahier des charges**. Dans le contexte de notre exemple, ces spécifications seront **très précises**, car elles déterminent la **légalité** de l'opération.

Dans les faits, il est **plus pertinent** de **rassembler des informations complémentaires** au **même endroit**. Vous aurez donc tout intérêt à **compléter vos descriptions** au sein même de vos **spécifications fonctionnelles** pour **mieux contextualiser** les attendus techniques.

En ayant **défini et justifié vos choix technologiques**, **légitimé** votre **solution socle** et apporté toutes les **précisions utiles** sur vos **spécifications** fonctionnelles, vous aurez **communiqué les informations nécessaires** à vos équipes de développement pour **concrétiser** votre solution.

## Exercice : Quiz

[solution n°1 p.23]

### Question 1

Voici plusieurs affirmations à propos du Cahier Des Charges Technique (CDCT). Identifiez celles qui sont exactes :

- ☐ Le CDCT peut parfois constituer un livrable indépendant, séparé du CDCF.
- ☐ Le CDCT présente les attendus en termes d'usages.
- ☐ Le CDCT présente les technologies sélectionnées dans le cadre du projet.
- ☐ Le CDCT peut influencer les choix créatifs ou fonctionnels.
- ☐ Le CDCT présente l'organisation générale du projet notamment au travers de l'équipe et de la gouvernance.

### Question 2

Que pourrait contenir votre présentation des grands axes technologiques dans votre cahier des charges ?

- ☐ La justification du choix d'un développement sur-mesure.
- ☐ La présentation des langages socles pour les développements.
- ☐ Un comparatif des systèmes de gestion de contenus potentiels.
- ☐ L'identification d'un framework qui sera utilisé par les développeurs.
- ☐ La justification du choix d'un développement hybride pour une application mobile.
- ☐ L'identification d'une extension de CMS pour développer une fonctionnalité.

### Question 3

« Dans une équipe projet, c'est au **chef de projet**, qui va rédiger le cahier des charges, de **déterminer les choix technologiques**. » **Vrai ou faux ?**

- ☐ Vrai
- ☐ Faux

#### Question 4

Dans votre cahier des charges technique, le **schéma d'architecture technique** fait écho à un **outil de représentation** utilisé dans le cahier des charges **fonctionnel**. Lequel ?

- ☐ Le périmètre fonctionnel
- ☐ La metro-map
- ☐ L'écosystème digital
- ☐ Les fiches *personas*

#### Question 5

Quelle est la façon **la plus efficace** de présenter les **spécifications techniques** nécessaires au développement de **chaque fonctionnalité** prévue dans le projet ?

- ☐ Compléter les spécifications fonctionnelles existantes par un schéma d'architecture.
- ☐ Dupliquer les spécifications fonctionnelles et remplacer les descriptions de chaque fonctionnalité par les informations techniques associées.
- ☐ Compléter les spécifications fonctionnelles existantes et y ajouter les informations techniques associées.
- ☐ Dupliquer les spécifications fonctionnelles existantes et y ajouter des colonnes pour préciser la couverture fonctionnelle.

### III. Préciser les spécificités de la réalisation

Votre **cahier des charges technique** a à ce stade **précisé votre solution** du point de vue de sa **conception technologique**. Vous pourrez le compléter en **ajoutant des instructions** ou en **détaillant des processus** pour préciser **la manière** de la réaliser.

**Après le « quoi », vous détaillerez donc le « comment », mais toujours dans une perspective de développements.**

#### A. Présenter l'environnement de développement

Traditionnellement, l'**environnement de travail** pour un **projet digital** se compose de **trois espaces bien distincts** qui permettent chacun d'accomplir des **tâches spécifiques** : l'environnement de **développement**, de **préproduction** et de **production**.

##### 1. Environnement de développement, pour produire

L'**environnement de développement** en tant que tel, c'est celui **dans lequel les équipes vont construire leur code**. Chaque développeur intervient depuis sa machine et **dépose ses fichiers** dans un **espace spécifique** où **tous les membres** de l'équipe technique peuvent intervenir.

Cet espace est **configuré** pour **répondre aux besoins des développeurs**, leur permet une grande **réactivité**, une bonne **gestion des versions** avec la possibilité de **revenir en arrière** en cas de problème. Il les aide à **collaborer** et à **documenter** la réalisation **au fur et à mesure** de sa production.

## 2. Environnement de préproduction, pour tester

Une fois les développements (ou une partie seulement) réalisés, ils sont **déposés dans un autre environnement : la préproduction**. C'est **un autre espace, séparé** de l'environnement de développement, qui **ressemble en tous points** à celui dans lequel on livrera **le produit définitif**.

C'est dans cet espace qu'on va pouvoir **réaliser les tests**, et en particulier **les tests fonctionnels**. Le **chef de projet** et le commanditaire pourront **se connecter** à cet environnement pour effectuer leur **recette**.

**Inaccessible aux utilisateurs**, il permet de **tester ses réalisations** dans un **environnement réaliste**, sans pour autant **risquer d'abîmer la solution déjà en place**. La **mise à jour** d'une fonctionnalité ou **l'ajout d'un nouveau développement** pourra générer des **anomalies** sur une solution déjà existante. Dans ce cas, on parlera « **d'effets de bord** » ou de « **régressions** ». On pourra donc **les constater** et **les corriger** avant de rendre la réalisation disponible pour les utilisateurs.

## 3. Environnement de production, pour livrer

Le troisième environnement à considérer, c'est **celui dans lequel on placera la réalisation finale**. Il accueillera **le résultat du projet**. On parlera **d'environnement de production**, et c'est pour cela qu'un des derniers événements de votre projet sera la « **mise en production** » (ou **MEP**, pour les intimes).

**On n'intervient généralement pas directement dans un environnement de production**, puisque c'est celui **dans lequel évoluent les utilisateurs**. Il doit donc rester aussi **stable** que possible.

### Exemple

La **définition d'environnement séparé** est **essentielle** pour **sécuriser** vos interventions techniques.

Imaginez par exemple que vous vous contentiez d'**un seul espace pour tester et délivrer** votre solution. Le jour où vous souhaitez **mettre en place une évolution**, tout fonctionne **parfaitement** sur la **machine du développeur** (son environnement de **développement**).

Vous décidez alors de **faire une mise en ligne** (mise en **production**) pour **déployer votre évolution**. Mais **une fois la nouvelle fonctionnalité sur le site**, vous vous apercevez que **plus rien ne fonctionne** : l'ajout **génère une erreur** avec la base de données et affiche un message cabalistique, quelle que soit la page que vous tentez de consulter.

**Tout le temps que votre développeur passera à corriger cette anomalie**, votre site **ne sera plus accessible**, ce qui pourrait avoir des **conséquences importantes** sur votre activité.

**Ces trois environnements** seront donc **décrits dans le cahier des charges technique**. L'**environnement de développement** sera **propre à votre structure**. Généralement la **préproduction** et la **production** sont gérées au niveau de **l'hébergeur** afin de s'assurer que les deux espaces soient bien **identiques**.

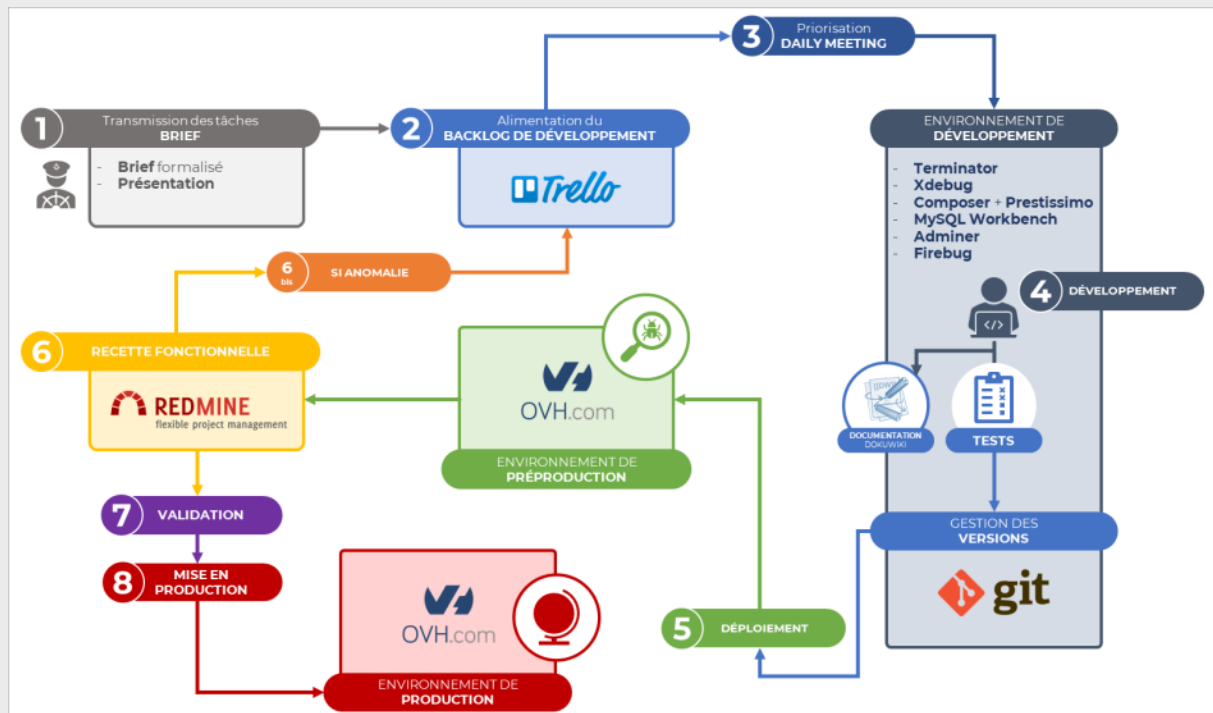
## 4. Détail de l'environnement de développement

L'**ensemble des processus** mis en place **dans le cadre des développements**, les **outils** que vos équipes techniques vont utiliser ainsi que les **supports et langages** utilisés seront détaillés dans la **présentation de votre environnement de développement**.

Comme pour beaucoup d'autres représentations dans votre cahier des charges, **un schéma** sera tout à fait **adapté** pour **partager rapidement les informations clés**.

### Exemple

Voici **un exemple d'environnement de développement** que vous pourriez intégrer à votre cahier des charges :



Il reprend le **processus global de travail** pour assurer la **production technique**. Les **trois espaces de développement, préproduction et production** sont représentés.

L'**environnement de développement** en tant que tel est **détaillé** en précisant les **outils utilisés** par les développeurs, la plateforme de **documentation**, la présence de procédure de **tests** et l'utilisation du **système de gestion de versions Git**.

C'est une **représentation claire** qui permet de **partager outils et processus**.

## B. Lister les contraintes et procédures techniques

Le cahier des charges technique **s'adresse en grande partie aux équipes de réalisation**. Vous allez donc y lister **l'ensemble des consignes et contraintes** applicables dans le cadre de votre projet.

### 1. Contraintes de structuration et bonnes pratiques

Ces contraintes seront en grandes parties **liées aux choix technologiques** présentés plus haut, même s'il sera toujours utile de **rappeler les bonnes pratiques** de développement **transversales**.

Vous pourrez ainsi préciser par exemple que les développements seront réalisés « **de manière pérenne, accessible et maintenable** », ou encore que l'ensemble des développements sera **commenté** de façon **standardisée**.

### Exemple

Voici **quelques bonnes pratiques transversales** destinées aux **développeurs** qui pourraient être présentes dans votre cahier des charges technique :

- **Aucune donnée utilisateur n'est transmise en clair dans les URL.**
- **La vérification** des champs de formulaire s'effectue **côté client** sans générer de connexion au serveur.

- Les **champs de formulaire** sont associés à un **attribut « type » cohérent** avec la donnée attendue (date, email, number, password, tel, time, url).
- La **langue du contenu** présenté est **précisée dans le code** chaque page par l'**attribut « lang = "fr" »**.
- Le **fichier sitemap.xml** est généré automatiquement et reflète les contenus publiés.
- Etc.

Impliquez le responsable des équipes techniques afin de formaliser un référentiel de bonnes pratiques propre à votre structure que vous pourrez ensuite réutiliser pour l'ensemble de vos futurs projets. Votre cahier des charges technique pourra alors se contenter d'y faire référence et éventuellement le présenter en annexe.

Les noms de fichiers créés pour la réalisation, les noms des fonctions et des variables pourront être contraint par un certain formalisme. On parlera de « **nomenclature** ». Celle-ci instaurée, vous faciliterez le fonctionnement général de votre réalisation et permettrez à vos équipes de développement de travailler ensemble plus efficacement.

Il faut noter qu'une **nomenclature cohérente et rigoureuse** favorise également les tâches de **maintenance**.

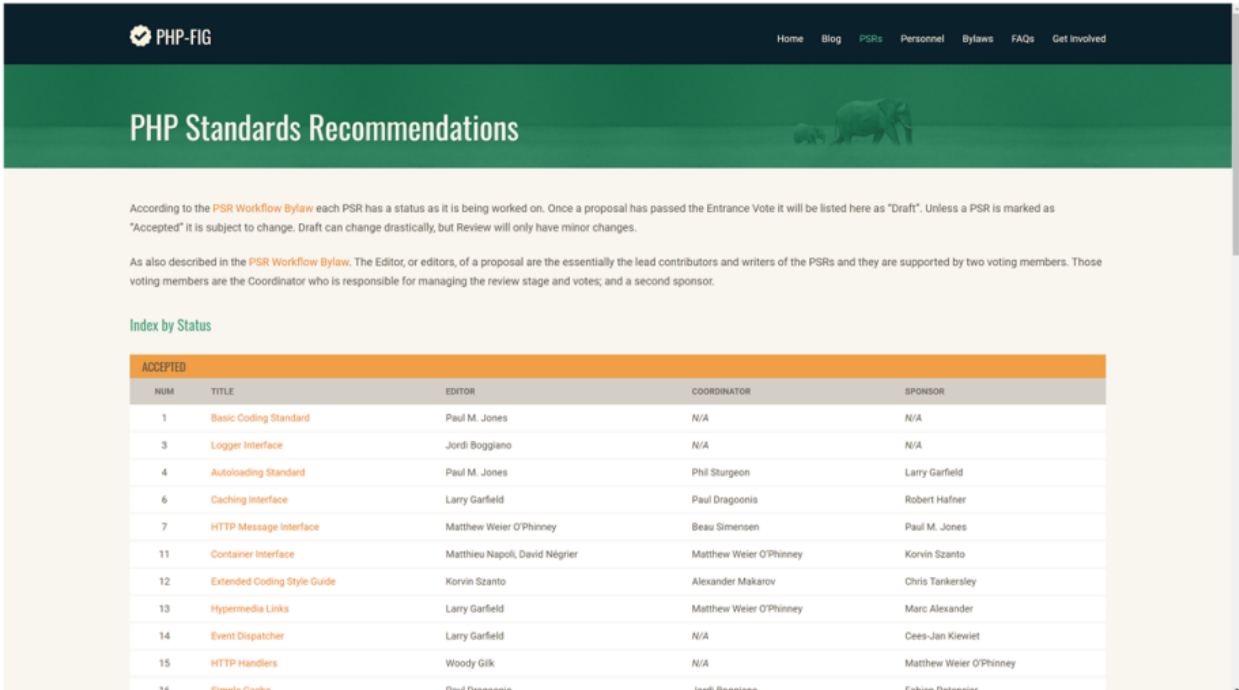
#### Exemple

Voici un exemple de recommandations à respecter pour définir des noms de fichiers :

- Pas de **caractères spéciaux** ni d'**accents**.
- Pas d'**espace**, séparation des termes par le **caractère de soulignement « \_ »**, le **tiret « - »** ou l'utilisation de **WikiWords**.
- L'utilisation de **l'anglais est privilégiée** dans les noms de fichiers destinés à un **traitement invisible** pour l'utilisateur.
- Les **fichiers HTML** correspondants à des **contenus statiques** doivent être **nommés en cohérence** avec les contenus. Dans ce cas, on se réfère aux **recommandations SEO** correspondant aux contenus affichés.
- Les noms de **fichiers**, de **variable** et de **fonctions** sont **explicites** dans le contexte. On évitera par exemple de nommer un fichier « user.php ». On préférera « user-login.php » ou « user-profil.php ».
- Etc.

Ce n'est **toujours pas une liste universelle** ; définissez **votre propre nomenclature** de façon cohérente avec **vos modes de fonctionnement** et vos **contraintes spécifiques** au projet, par exemple en termes d'**accessibilité**.

Pour **tous les langages de développement**, il existe des **conventions de codage**, généralement bien documentées. Vous pourrez alors utilement **les rappeler**. Par exemple, pour les développeurs qui codent en **PHP**, vous pouvez retrouver des **références en ligne** comme **php-fig.org<sup>1</sup>** (en anglais)



According to the **PSR Workflow Bylaw** each PSR has a status as it is being worked on. Once a proposal has passed the Entrance Vote it will be listed here as "Draft". Unless a PSR is marked as "Accepted" it is subject to change. Draft can change drastically, but Review will only have minor changes.

As also described in the **PSR Workflow Bylaw**. The Editor, or editors, of a proposal are the essentially the lead contributors and writers of the PSRs and they are supported by two voting members. Those voting members are the Coordinator who is responsible for managing the review stage and votes; and a second sponsor.

**Index by Status**

ACCEPTED				
NUM	TITLE	EDITOR	COORDINATOR	SPONSOR
1	Basic Coding Standard	Paul M. Jones	N/A	N/A
3	Logger Interface	Jordi Boggiano	N/A	N/A
4	Autoloading Standard	Paul M. Jones	Phil Sturgeon	Larry Garfield
6	Caching Interface	Larry Garfield	Paul Dragoonis	Robert Hafner
7	HTTP Message Interface	Matthew Weier O'Phinney	Beau Simensen	Paul M. Jones
11	Container Interface	Matthieu Napoli, David Négrier	Matthew Weier O'Phinney	Korvin Szanto
12	Extended Coding Style Guide	Korvin Szanto	Alexander Makarov	Chris Tankersley
13	Hypermedia Links	Larry Garfield	Matthew Weier O'Phinney	Marc Alexander
14	Event Dispatcher	Larry Garfield	N/A	Cees-Jan Kiewiet
15	HTTP Handlers	Woody Glik	N/A	Matthew Weier O'Phinney
16	Simple Cache	Paul Dragoonis	Jordi Boggiano	Fabien Potencier

Si vous devez respecter des **normes d'accessibilité** ou de **qualité Web**, vous devrez alors **préciser les référentiels correspondants** dans votre cahier des charges technique.

## 2. Modalités de livraison et d'archivage

Les **modalités de livraison et d'archivage** complètent les contraintes définies de façon plus générale dans **l'organisation projet** de votre cahier des charges fonctionnel.

Elles seront **liées à votre environnement de développement** présenté précédemment. Vous les définirez **selon votre contexte**, mais vous veillerez toujours à :

- Rappeler la nécessité de **vérifier les réalisations** avant d'effectuer une livraison, et éventuellement à mettre en place des **procédures automatiques de test**.
- Être en mesure de **retracer les livraisons** effectuées.
- **Formaliser les livraisons** au travers d'un **processus standardisé** pour gagner en **efficacité**.
- **Conserver les différentes versions** produites de manière à pouvoir **réutiliser une ancienne version** en cas de nécessité.

Les modalités de livraison vont souvent impliquer des **procédures techniques particulières** pour les équipes de développement, par exemple la nécessité de **mettre à la disposition de l'équipe entière** le code produit par un développeur sur son propre poste, et qu'il faudra préciser également.

<sup>1</sup> <https://www.php-fig.org/psr/>

## C. Présenter l'hébergement

**L'hébergement** est toujours une thématique à traiter dans un projet digital. En effet, les **fichiers**, les **données**, les informations utilisés pour les **statistiques** par exemple **doivent être stockés** pour pouvoir être **exploités**. Pour un projet Web tout particulièrement, l'hébergement sera **obligatoirement détaillé** dans le cahier des charges technique.

Dans certaines situations, vous serez amenés à gérer cet hébergement par vous-même, mais la plupart du temps, vous aurez **recours à un hébergeur** qui proposera, en contrepartie d'un abonnement annuel, non seulement **un espace de stockage**, mais également les **services nécessaires** pour assurer une **parfaite disponibilité** des contenus hébergés et la **maintenance technique** de **l'infrastructure**.

Dans votre cahier des charges techniques, vous préciserez les **caractéristiques attendues** pour l'hébergement nécessaire au **bon fonctionnement** de vos réalisations.

### Exemple

**Vos équipes techniques** peuvent vous **transmettre les caractéristiques de l'hébergement nécessaire** pour votre projet, notamment en ce qui concerne la **puissance** des machines, leur **configuration**, le **trafic** attendu ou les **capacités de stockage**.

Une fois ces éléments connus, **l'hébergeur vous communiquera les détails de la configuration** proposée. Vous pourrez alors **simplement l'intégrer** à votre cahier des charges technique :

#### Configuration

Pour répondre de façon optimale à votre demande, nous avons élaboré une infrastructure regroupant 2 instances avec les configurations suivantes :

- 1 serveur virtuel web, Dual Core 2.6 GHz, RAM 16 Go, HDD 80 Go.
- 1 serveur virtuel sql, Dual Core 2.6 GHz, RAM 16 Go, HDD 80 Go.
- 250 Go de trafic mensuel, sauvegarde incrémentale sur 7 jours, monitoring personnalisé et accès au support 24h/24 et 7j/7.

L'hébergeur garantit une disponibilité du réseau de l'hébergeur de 99,9%.  
En cas de non-respect de la disponibilité, l'hébergeur vous accordera un avoir sur votre prochaine facture dont le montant sera calculé en fonction de la durée d'indisponibilité du réseau de l'hébergeur :

Disponibilité	Avoir*
< 0,9990	3%
< 0,9950	5%
< 0,9850	10%
< 0,9750	15%
< 0,9700	20%
< 0,9500	40%
< 0,9000	70%
< 0,850	100%

\*Avoir de la redevance mensuelle HT du mois concerné par l'indisponibilité.  
Toutes interventions de l'hébergeur sur les serveurs et le réseau de l'hébergeur telles que les opérations de maintenance préventive, corrective ou évolutive et les tests associés dont le Client aura été préalablement informé ne sont pas prises en compte dans le calcul de l'indisponibilité des Services.

**L'hébergeur garantit une disponibilité du réseau de l'hébergeur de 99,9 %**

Configuration	Statut
Location de l'infrastructure	✓
Accès au réseau burstable	Jusqu'à 1 Gbps non bridé
GTI (Garantie de Temps d'Intervention)	30 minutes
GTR (Garantie de Temps de Rétablissement)	2 heures
Trafic mensuel inclus par serveur	250 Go
Sauvegarde sur site distant	Incrémentale quotidienne 7 jours
Equipement réseaux et électriques doublés	✓
Baies équipées de 2 switches	✓
Monitoring personnalisé	✓
Accès au support 24h/24 et 7j/7	✓

Notez qu'au-delà des **descriptions précises de l'infrastructure et des serveurs** mis à disposition, un hébergeur s'engage généralement sur un **niveau de disponibilité de son réseau**. C'est une **information importante** à faire figurer dans votre cahier des charges technique qui vous permettra de **rassurer sur la fiabilité** de votre solution dans sa phase de vie.

Soyez aussi attentifs à **l'accès au support technique** proposé par l'hébergeur, votre cahier des charges pourra même **indiquer les moyens de contacter ce service**.

La **logique de sauvegarde de vos données hébergées** est par ailleurs importante à connaître et à **préciser**, car elle constitue un moyen de **revenir rapidement à un système opérationnel** en cas de gros problème sur votre solution.



Notez que **pour certains projets** vous devrez **tenir compte de contraintes spécifiques**. Si vous prévoyez de manipuler des **données financières**, il vous faudra sans doute trouver un hébergeur ayant obtenu la **certification PCI-DSS**. De la même façon, si vous déployez une application dans un cadre médical, vous devrez peut-être trouver une solution d'hébergement disposant d'un **agrément HDS** (pour **Hébergement des Données de Santé**). Le **contexte général** de votre projet pourra **influencer fortement vos choix** en termes d'hébergement.

## D. Anticiper la maintenance et les évolutions

Une solution digitale est amenée à **évoluer dans le temps**. Elle devra s'adapter aux **évolutions technologiques**, aux **usages** qui changent rapidement, actualiser son **interface**, pour répondre à des **standards changeants**. Par ailleurs, des **anomalies** peuvent toujours apparaître une fois le projet terminé. **Même si votre recette a été méticuleuse**, vous ne pourrez **jamais anticiper toutes les anomalies possibles** durant la **phase de vie** de la solution.

C'est pour cela que beaucoup de prestataires proposent des **prestations de maintenance** sur les solutions qu'ils ont développées. Notons au passage que cette maintenance peut tout aussi bien être effectuée **par le prestataire** qui a développé la solution que **par une autre société**.

On parle généralement de **TMA**, pour *Tierce Maintenance Applicative*. Elle peut être de **deux types** :

- La **TMA corrective** a pour but de **résoudre les anomalies** qui peuvent survenir **en cours d'exploitation** de la solution.

Pour cela, les équipes qui vont intervenir doivent **parfaitement connaître le produit** de façon à **pouvoir agir efficacement**. Ce sera d'autant plus important que, souvent, les **contrats de maintenance** incluent un « **SLA** » pour *Service-Level Agreement*. Sans entrer dans les détails, cet **engagement de niveau de service** vous contraint à une **prise en compte de l'anomalie** dans un **délai donné (généralement très court)** et peut parfois imposer également des **délais de résolution**, quitte à revenir à **une version fonctionnelle antérieure** à celle qui a généré l'anomalie.

Cela nécessite donc, pour le prestataire qui a réalisé la solution, d'en **préserver la connaissance et la maîtrise** au sein de ses équipes. C'est pour cela que **le montant** d'une TMA peut parfois être **très élevé**.

Si la maintenance est **assurée par une autre société**, le prestataire aura dû **fournir toutes les informations utiles** pour lui permettre d'intervenir sur la solution.

- La **TMA évolutive** est une **autre forme** de maintenance. Elle ne vise pas à corriger des anomalies, mais plutôt à **faire évoluer la solution**.

Elle prend en charge des **modifications mineures** sur le produit de manière à en **améliorer les performances** ou pour **ajouter des fonctionnalités nouvelles** qui viendront augmenter l'intérêt ou l'attrait de la solution.

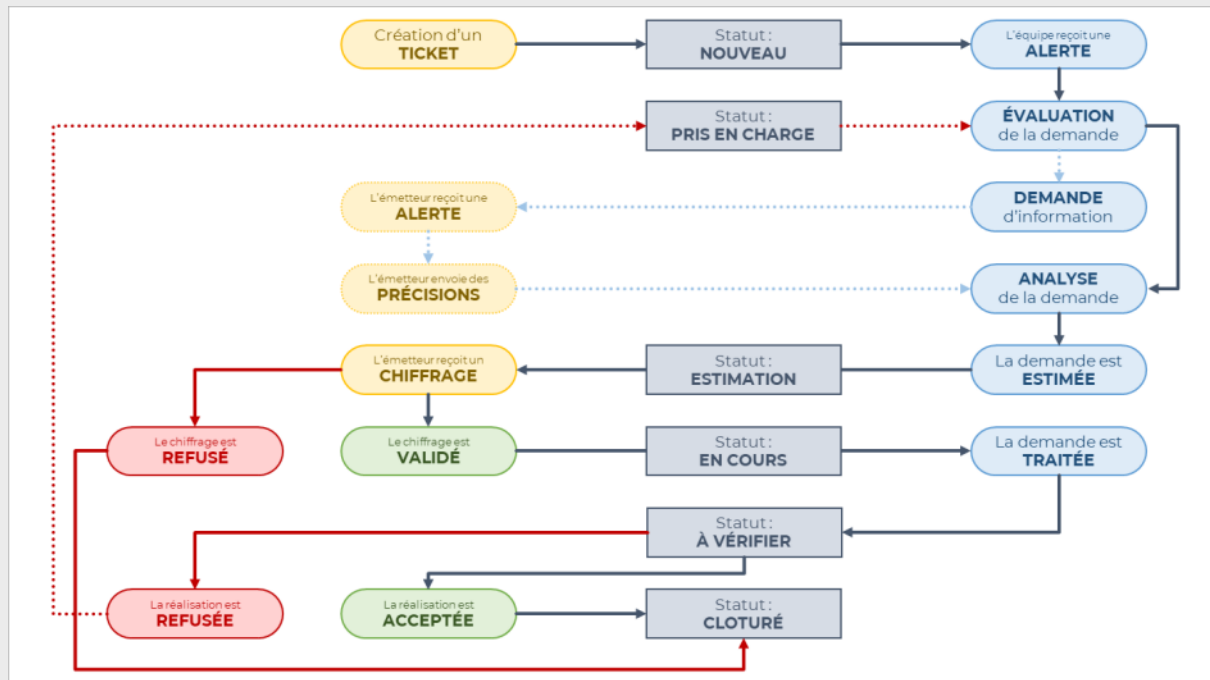
Ces **demandes de correction d'anomalies ou d'évolutions** feront l'objet de **processus spécifiques** que votre cahier des charges technique **devra détailler**.

### Exemple

Votre cahier des charges technique pourra intégrer votre **processus de maintenance**. Généralement, les **demandes de maintenance**, qu'il s'agisse de **signaler une anomalie** ou de **demande une évolution**, reposent sur un **système de gestion de tickets** (via **Jira** ou **Redmine** par exemple).



Vous pourrez alors le partager en réalisant ce type de **schéma** :



Le processus ainsi présenté établit clairement les étapes de traitement de la demande et les actions attendues.

Les actions menées **dans le cadre d'une maintenance évolutive** sont généralement de **faible envergure**. Elles peuvent être **traitées rapidement** par l'équipe en place **sans avoir à reconstruire une organisation et une équipe dédiée**.

Si la modification attendue **implique un volume de charge important**, une **nouvelle conception** et des **créations complémentaires** par exemple, alors il sera souvent préférable de **basculer en mode projet** et d'envisager une **refonte**.

**Quel que soit le type de maintenance**, vous voyez que **la documentation** de votre projet revêt un **caractère essentiel**. Si votre réalisation n'est pas documentée, alors il sera très difficile d'en corriger les anomalies qui pourraient surgir ou de la faire évoluer.

En synthèse, **retenez bien ces informations clés** lorsque vous rédigerez votre cahier des charges technique :

- Les **processus de maintenance** doivent être **anticipés** et **décrits** dans le cahier des charges technique.
- La **documentation technique** est **indispensable** pour permettre cette maintenance au cours du temps, et vous devrez **en indiquer les modalités** dans votre cahier des charges technique.

En anticipant **comment** vous pourrez **maintenir** et **faire évoluer** votre solution **dans sa phase de vie**, vous mettez le **point final** à votre **cahier des charges technique**.

## Exercice : Quiz

[solution n°2 p.24]

### Question 1

Quels sont les **trois environnements** que vous devrez décrire dans votre cahier des charges technique ?

- ☐ L'environnement de développement
- ☐ L'environnement local

- ☐ L'environnement d'hébergement
- ☐ L'environnement de production
- ☐ L'environnement de préproduction
- ☐ L'environnement de post-production

Question 2

« Les mots dans les noms de fichiers doivent être **séparés par des tirets** « - » ou des caractères de soulignement « \_ ». » Dans quelle **catégorie** pourriez-vous classer cette **contrainte** extraite d'un cahier des charges technique ?

- ☐ Bonne pratique de développement
- ☐ Nomenclature
- ☐ Conventions de codage
- ☐ Référentiel

Question 3

Votre projet comportera le traitement et le stockage de **données de santé**. Quels **types d'informations** seront utiles dans votre cahier des charges technique lorsque vous décrierez **l'hébergement** proposé par un prestataire externe ?

- ☐ La configuration des machines serveur et la capacité de stockage
- ☐ Le volume de trafic mensuel
- ☐ Le système de gestion de versions (Git par exemple)
- ☐ Les modalités de sauvegarde automatique des données
- ☐ Le niveau de disponibilité du réseau
- ☐ L'agrément HDS
- ☐ La certification PCI-DSS

Question 4

C'est un **environnement destiné aux développeurs** qui leur permet de **collaborer** pour construire la solution attendue, de la documenter, de gérer et d'archiver les **différentes versions** du code. De quel environnement s'agit-il ?

- ☐ L'environnement de développement
- ☐ L'environnement local
- ☐ L'environnement d'hébergement
- ☐ L'environnement de production
- ☐ L'environnement de préproduction

Question 5

Voici plusieurs affirmations à propos de la **TMA évolutive**. Indiquez celles qui sont exactes :

- ☐ Permet de résoudre les anomalies qui peuvent survenir en phase de vie.
- ☐ Permet de réaliser des modifications mineures pour améliorer les performances de la solution.
- ☐ Permet d'ajouter des fonctionnalités nouvelles à un produit existant.
- ☐ Permet de prendre en charge des évolutions lourdes nécessitant plusieurs semaines de travail.
- ☐ Implique la mise en place d'un processus spécifique pour le traitement des demandes, par exemple via un système de gestion de tickets.

## V. Essentiel

### Fondamental

Le **cahier des charges technique** est un document qui **viendra compléter votre cahier des charges fonctionnel**. Il pourra être **intégré à ce dernier** ou faire l'objet d'un **livrable séparé**. Il a pour but de **décrire votre solution** sous un **angle technique**. Avec le cahier des charges fonctionnel, il partage **la nécessité d'expliquer et de justifier les choix** effectués. Vous y aborderez la **présentation de vos grands axes de développement**, notamment en partageant votre **stratégie de réalisation** : développer un site **à partir de rien**, vous appuyez sur une **solution existante** ou encore opter pour **un mélange** de ces deux options.

Si vous vous appuyez sur **une solution CMS existante**, vous devrez alors **la présenter** et en **mettre en lumière les caractéristiques**. L'utilisation d'un **comparatif chiffré** pourra vous y aider. Un des **arguments importants** à présenter sera la **couverture fonctionnelle** de la solution technique retenue. Il vous faudra indiquer **la part de fonctionnalités** qui pourra être **couverte nativement par votre CMS** et celle qu'il vous faudra **développer**.

Le **schéma d'architecture technique** vous aidera à **partager une vue d'ensemble** de votre future réalisation. Il reflète la **dimension technologique** de votre **écosystème digital**, en indiquant les **flux de données** et les **interactions** entre les plateformes technologiques concernées par le projet. Si nécessaire, vous devrez **compléter vos spécifications fonctionnelles** par des **informations indispensables** aux développeurs.

La **manière** dont vous allez **réaliser les développements** devra aussi être indiquée dans le cahier des charges technique. Vous y détaillerez notamment les **caractéristiques de votre environnement de développement**. Les **outils** et les processus mobilisés constituent des informations propres à **rassurer le commanditaire** tout en étant essentielles pour **guider les équipes** de production. La précision de **bonnes pratiques** de développement, d'une **nomenclature** ou la référence à des **normes et référentiels** seront tout aussi importants.

L'**hébergement** sera également **utile à préciser**, tout comme la **manière** dont vous envisagez la **maintenance** et les **évolutions** de votre future réalisation. Si le cahier des charges technique ne constitue **pas la documentation** de votre projet, c'est à la fois **un cadre** et **un guide** pour en **optimiser les développements**.

## VI. Auto-évaluation

### Exercice 1 : Quiz

[solution n°3 p.26]

#### Question 1

« Dans un cahier des charges technique, chaque **choix technologique** doit être **justifié**. » Que pensez-vous de cette affirmation ?

- ☐ C'est vrai, il est important de faire comprendre pourquoi une solution a été retenue.
- ☐ C'est faux, certains choix s'imposent d'eux-mêmes et n'ont pas besoin d'être justifiés.
- ☐ C'est faux, le CDCT fait la synthèse des solutions retenues, les justifications sont inutiles.

#### Question 2

« Les environnements de **préproduction** et de **production** devraient avoir une configuration technique **identique**. »  
**Vrai ou faux ?**

- ☐ Vrai
- ☐ Faux

Question 3

Il présente un **processus** de travail, les **outils** et logiciels utilisés par l'équipe de développement, les principes de **documentation** ou encore le système de **gestion de version**. De quel **outil visuel** du cahier des charges technique parle-t-on ici ?

- ☐ Le schéma d'architecture
- ☐ Le tableau de couverture fonctionnelle
- ☐ Le schéma de l'environnement de développement
- ☐ Le schéma du processus de maintenance

Question 4

On y dépose les **fichiers** réalisés par les développeurs afin de pouvoir organiser les **tests fonctionnels** (la recette) de l'application. De quel environnement s'agit-il ?

- ☐ Développement
- ☐ Préproduction
- ☐ Production

Question 2

« La **couverture fonctionnelle** désigne le volume de fonctionnalités nativement proposées par un CMS. Plus elle est importante, plus le CMS choisi sera **adapté au projet**. » **Vrai ou faux ?**

- ☐ Vrai
- ☐ Faux

**Exercice 7**

[solution n°4 p.28]

Dès lors que vous appréhendez un projet Web, vous devrez, dans votre cahier des charges technique, justifier votre approche de développement. Se posera alors rapidement la question de savoir si vous allez opter pour une solution CMS existante que vous personnaliserez, ou pour un développement sur-mesure. Dans les deux cas, vous devrez trouver des arguments pour justifier vos positions.

Plusieurs arguments vous sont proposés ci-dessous. Identifiez ceux qui justifient un développement basé sur un CMS et ceux qui plaident en faveur d'un développement sur-mesure.

Proposer une réalisation moins vulnérable aux attaques potentielles.

Proposer une interface d'administration très simplifiée.

Profiter d'une interface d'administration déjà conçue.

Garder la maîtrise sur le code développé.

Bénéficier d'une communauté de développeurs établie et de nombreuses ressources en ligne.

Proposer une solution connue qui pourra être maintenue par un tiers.

Réduire significativement les temps de développement.

Implémenter des fonctionnalités très spécifiques propres au projet.

Éviter du code inutile correspondant à des fonctionnalités non utilisées.

Profiter de mise à jour rapide en cas de découverte d'une faille de sécurité.

Pour justifier une solution basée sur un CMS	Pour justifier une solution développée sur-mesure

## Exercice 8

[solution n°5 p.28]

Dans le cahier des charges technique, il sera important de décrire l'environnement global dans lequel évolueront les équipes techniques, mais également de détailler les trois environnements spécifiques dans lesquels les développeurs (et les autres acteurs du projet) vont évoluer. Encore faut-il bien en identifier les caractéristiques.

Complétez les phrases ci-dessous en utilisant les bons termes parmi : développement, préproduction et production.


1. L'environnement de [ ] est l'espace dans lequel les développeurs construisent leur code.
2. Lorsque les développements sont réalisés, ils seront placés dans l'environnement de [ ] afin que la recette fonctionnelle puisse avoir lieu.
3. Lors d'une mise à jour, on peut constater dans l'environnement de [ ] des effets de bord ou des régressions, qu'on pourra corriger sans affecter la version en ligne.
4. Lorsqu'on effectue la livraison finale du projet, on parle de « mise en [ ] », parce qu'on place les fichiers définitifs dans l'environnement de [ ].
5. Une « mise en [ ] » précède la phase de recette fonctionnelle.
6. Les environnements de [ ] et de [ ] doivent être deux environnements séparés, mais de configuration identique.
7. Avant de livrer en [ ], les développeurs devraient faire des tests sur leur poste, dans leur environnement de [ ].
8. La documentation technique de la réalisation se produit dans l'environnement de [ ].

## Solutions des exercices




**Exercice p. 9 Solution n°1****Question 1**

Voici plusieurs affirmations à propos du Cahier Des Charges Technique (CDCT). Identifiez celles qui sont exactes :

- ☒ Le CDCT peut parfois constituer un livrable indépendant, séparé du CDCF.
  - ☐ Le CDCT présente les attendus en termes d'usages.
  - ☒ Le CDCT présente les technologies sélectionnées dans le cadre du projet.
  - ☒ Le CDCT peut influencer les choix créatifs ou fonctionnels.
  - ☐ Le CDCT présente l'organisation générale du projet notamment au travers de l'équipe et de la gouvernance.
-  Le Cahier Des Charges Technique (CDCT) peut constituer un livrable indépendant du Cahier Des Charges Fonctionnel (CDCF). Si ce dernier présente les attendus en termes d'usages ainsi que l'organisation générale du projet, il est avant tout utile pour présenter la solution sous l'angle fonctionnel. Le CDCT, lui, présente notamment les technologies sélectionnées pour la mise en œuvre du projet. En cela, il pourra influencer, voire même contraindre, certains choix créatifs ou fonctionnels.

**Question 2**

Que pourrait contenir votre présentation des grands axes technologiques dans votre cahier des charges ?

- ☒ La justification du choix d'un développement sur-mesure.
  - ☒ La présentation des langages socles pour les développements.
  - ☐ Un comparatif des systèmes de gestion de contenus potentiels.
  - ☒ L'identification d'un framework qui sera utilisé par les développeurs.
  - ☒ La justification du choix d'un développement hybride pour une application mobile.
  - ☐ L'identification d'une extension de CMS pour développer une fonctionnalité.
-  Lorsque vous présentez les grands axes technologiques de votre solution, vous partagez des choix directeurs : un développement sur-mesure ou le recours à une solution de type CMS, les langages socles ou les frameworks utilisés pour les développements, le choix d'un développement hybride ou natif pour une application mobile, etc. Si vous optez pour un CMS (ce que vous justifierez donc dans la présentation de vos grands axes), vous pourrez bien entendu présenter un comparatif des solutions possibles, mais vous le ferez au moment de décrire la solution retenue, pas dans la présentation des grands axes. De même, vous décrierez les extensions de CMS utilisées au moment de la présentation de la couverture fonctionnelle. Les grands axes technologiques justifient une approche globale de développement ; on n'entre pas encore dans les détails à ce stade.

**Question 3**

« Dans une équipe projet, c'est au **chef de projet**, qui va rédiger le cahier des charges, de **déterminer les choix technologiques**. » **Vrai ou faux ?**

- ☐ Vrai
- ☒ Faux

- Q Le chef de projet sera garant de la bonne réalisation et du formalisme du cahier des charges, incluant les parties fonctionnelles et techniques. Toutefois, le plus souvent, il n'aura pas la capacité de déterminer les choix technologiques. Les ressources spécialisées, comme un lead développeur, un directeur technique ou un chef de projet technique, définiront les technologies à utiliser. Le chef de projet pourra être amené à justifier ces choix, mais ils seront faits et argumentés au préalable par les équipes techniques.

#### Question 4

Dans votre cahier des charges technique, le **schéma d'architecture technique** fait écho à un **outil de représentation** utilisé dans le cahier des charges **fonctionnel**. Lequel ?

- ☐ Le périmètre fonctionnel
- ☐ La metro-map
- ☒ L'écosystème digital
- ☐ Les fiches *personas*

- Q L'architecture technique représente les échanges prévus entre les différentes plateformes réalisées dans le cadre du projet (les différents chantiers) et éventuellement avec les autres plateformes du système d'information ou des systèmes externes. Il sera donc naturellement en lien avec l'écosystème digital qui, lui, matérialise les interactions entre la solution socle et les autres plateformes digitales concernées par le projet. L'écosystème digital offre une vue globale du point de vue de l'utilisateur, le schéma d'architecture présente cette vue d'ensemble du point de vue technique.

#### Question 5

Quelle est la façon **la plus efficace** de présenter les **spécifications techniques** nécessaires au développement de **chaque fonctionnalité** prévue dans le projet ?

- ☐ Compléter les spécifications fonctionnelles existantes par un schéma d'architecture.
- ☐ Dupliquer les spécifications fonctionnelles et remplacer les descriptions de chaque fonctionnalité par les informations techniques associées.
- ☒ Compléter les spécifications fonctionnelles existantes et y ajouter les informations techniques associées.
- ☐ Dupliquer les spécifications fonctionnelles existantes et y ajouter des colonnes pour préciser la couverture fonctionnelle.

- Q Les spécifications fonctionnelles décrivent le comportement de chaque fonctionnalité attendue, du point de vue de l'utilisateur. Le cahier des charges technique doit fournir les indications nécessaires pour pouvoir les développer. Vous gagnerez donc en clarté (et beaucoup de temps) en complétant les spécifications existantes par les informations techniques requises. Dupliquer ces spécifications augmentera inutilement le volume de votre cahier des charges, et si vous supprimer les indications fonctionnelles pour les remplacer par des instructions techniques, il sera alors très compliqué de comprendre les attendus. Par ailleurs, le schéma d'architecture présente une vue d'ensemble du projet, mais ne décrit pas les fonctionnalités. La couverture fonctionnelle sera, elle, précisée au niveau du périmètre fonctionnel.

### Exercice p. 17 Solution n°2

#### Question 1

Quels sont les **trois environnements** que vous devrez décrire dans votre cahier des charges technique ?

- ☒ L'environnement de développement
- ☐ L'environnement local



- ☐ L'environnement d'hébergement
- ☒ L'environnement de production
- ☒ L'environnement de préproduction
- ☐ L'environnement de post-production

Q Votre cahier des charges devra détailler l'environnement de développement, au sein duquel vos équipes techniques vont réaliser leur travail. Vous décrirez également l'environnement de préproduction, destiné à la recette pour vérifier les réalisations, ainsi que l'environnement de production, qui est l'espace « *final* » dans lequel votre solution digitale sera livrée. Idéalement, ces trois espaces seront bien séparés.

### Question 2

« Les mots dans les noms de fichiers doivent être **séparés par des tirets** « - » ou des caractères de soulignement « \_ ». » Dans quelle **catégorie** pourriez-vous classer cette **contrainte** extraite d'un cahier des charges technique ?

- ☐ Bonne pratique de développement
- ☒ Nomenclature
- ☐ Conventions de codage
- ☐ Référentiel

Q Toutes les propositions présentées ici peuvent constituer des contraintes que les développeurs devront respecter. Toutefois, la manière de formaliser des noms de fichiers fait référence à une nomenclature. Les bonnes pratiques sont transversales et peuvent être établies au niveau de votre organisation. Les conventions de codage sont propres à un langage en particulier et les référentiels vont adresser des domaines transversaux tels que la qualité Web ou l'accessibilité.

### Question 3

Votre projet comportera le traitement et le stockage de **données de santé**. Quels **types d'informations** seront utiles dans votre cahier des charges technique lorsque vous décrirez **l'hébergement** proposé par un prestataire externe ?

- ☒ La configuration des machines serveur et la capacité de stockage
- ☒ Le volume de trafic mensuel
- ☐ Le système de gestion de versions (Git par exemple)
- ☒ Les modalités de sauvegarde automatique des données
- ☒ Le niveau de disponibilité du réseau
- ☒ L'agrément HDS
- ☐ La certification PCI-DSS

Q La présentation de l'hébergement dans le cahier des charges techniques doit donner toutes les informations utiles aux équipes techniques et celles nécessaires pour rassurer le commanditaire du projet. Ainsi, la description des configurations des machines serveurs, le niveau de disponibilité des infrastructures, les systèmes de sauvegarde en place ou le trafic mensuel accordé sont des informations précieuses. Dans le cadre de cette question, il vous faudra manipuler des données de santé ; un agrément HDS (Hébergement des Données de Santé) sera donc requis. Le système de gestion de version (Git) n'est pas lié à l'hébergement, mais à l'environnement de développement.

#### Question 4

C'est un **environnement destiné aux développeurs** qui leur permet de **collaborer** pour construire la solution attendue, de la documenter, de gérer et d'archiver les **différentes versions** du code. De quel environnement s'agit-il ?

- ☒ L'environnement de développement
- ☐ L'environnement local
- ☐ L'environnement d'hébergement
- ☐ L'environnement de production
- ☐ L'environnement de préproduction

**Q** C'est dans l'environnement de développement que les développeurs interviendront au quotidien. Dans cet espace, dont l'organisation sera décrite dans le cahier des charges technique, ils pourront collaborer, développer et documenter la solution réalisée dans le cadre du projet. Une fois leurs tâches effectuées et testées, ils pourront déposer les fichiers en préproduction afin de soumettre la réalisation à une recette fonctionnelle.

#### Question 5

Voici plusieurs affirmations à propos de la **TMA évolutive**. Indiquez celles qui sont exactes :

- ☐ Permet de résoudre les anomalies qui peuvent survenir en phase de vie.
- ☒ Permet de réaliser des modifications mineures pour améliorer les performances de la solution.
- ☒ Permet d'ajouter des fonctionnalités nouvelles à un produit existant.
- ☐ Permet de prendre en charge des évolutions lourdes nécessitant plusieurs semaines de travail.
- ☒ Implique la mise en place d'un processus spécifique pour le traitement des demandes, par exemple via un système de gestion de tickets.

**Q** La TMA évolutive a pour but de faire évoluer la solution réalisée dans le cadre du projet. La TMA corrective permettra, elle, de résoudre des anomalies qui pourraient survenir en cours d'exploitation de la solution. On utilisera la TMA évolutive pour réaliser des modifications mineures ou ajouter des fonctionnalités nouvelles. Pour fluidifier les interactions, on établira un processus spécifique qui sera avantageusement détaillé dans le cahier des charges technique. En revanche, on n'aura pas recours à la TMA pour prendre en charge des évolutions très lourdes : celles-ci nécessiteront de revenir en mode projet et de mobiliser une nouvelle équipe dédiée.

### Exercice p. 19 Solution n°3

#### Question 1

« Dans un cahier des charges technique, chaque **choix technologique** doit être **justifié**. » Que pensez-vous de cette affirmation ?

- ☒ C'est vrai, il est important de faire comprendre pourquoi une solution a été retenue.
- ☐ C'est faux, certains choix s'imposent d'eux-mêmes et n'ont pas besoin d'être justifiés.
- ☐ C'est faux, le CDCT fait la synthèse des solutions retenues, les justifications sont inutiles.

- Q Tout comme vous le ferez dans la partie fonctionnelle de votre cahier des charges, il sera important de justifier l'ensemble de vos choix technologiques. Le commanditaire, comme l'ensemble de l'équipe de réalisation, pourra ainsi comprendre ce qui a motivé les choix technologiques effectués, et donc, mieux les accepter. L'argumentation et la justification sont essentielles dans un cahier des charges.

### Question 2

« Les environnements de **préproduction** et de **production** devraient avoir une configuration technique **identique**. »

**Vrai ou faux ?**

☒ Vrai

☐ Faux

- Q C'est vrai : pour permettre une recette efficace et s'assurer que le résultat du projet sera bien fonctionnel en situation réelle, l'environnement de recette devrait présenter une configuration identique à l'environnement de production. Il sera toujours utile de donner cette précision dans le cahier des charges pour sécuriser les conditions de la recette.

### Question 3

Il présente un **processus** de travail, les **outils** et logiciels utilisés par l'équipe de développement, les principes de **documentation** ou encore le système de **gestion de version**. De quel **outil visuel** du cahier des charges technique parle-t-on ici ?

☐ Le schéma d'architecture

☐ Le tableau de couverture fonctionnelle

☒ Le schéma de l'environnement de développement

☐ Le schéma du processus de maintenance

- Q L'énoncé de cette question fait référence au schéma de l'environnement de développement qui a vocation à partager le mode d'organisation des équipes techniques. Le schéma d'architecture s'oriente sur la construction globale de la solution et ses relations avec des systèmes tiers. Le tableau de couverture fonctionnelle permet de bien visualiser les fonctionnalités spécifiques à développer, et le schéma du processus de maintenance se bornera à détailler les étapes du suivi d'une demande d'évolution. Ces quatre outils seront très utiles dans votre cahier des charges technique, mais chacun présentera des informations spécifiques.

### Question 4

On y dépose les **fichiers** réalisés par les développeurs afin de pouvoir organiser les **tests fonctionnels** (la recette) de l'application. De quel environnement s'agit-il ?

☐ Développement

☒ Préproduction

☐ Production

- Q Les productions des développeurs sont réalisées dans leur environnement de développement. Une fois finalisés, les fichiers sont déposés dans un environnement de recette : la préproduction. Ce n'est qu'une fois la recette effectuée et validée dans l'environnement de préproduction que l'ensemble des fichiers et des données seront copiés vers l'environnement de production lors de la « mise en production ». Cette dernière action constitue la livraison principale du projet.

### Question 2

« La **couverture fonctionnelle** désigne le volume de fonctionnalités nativement proposées par un CMS. Plus elle est importante, plus le CMS choisi sera **adapté au projet**. » **Vrai ou faux ?**

☒ Vrai

☐ Faux

Q La couverture fonctionnelle permet, à partir de votre périmètre fonctionnel, d'identifier l'ensemble des fonctionnalités portées nativement par une solution CMS. Si ces fonctionnalités n'ont pas besoin d'être développées en cours de projet, vous gagnerez un temps certain lors de la phase de production. Cela en fait un argument de poids pour sélectionner un système de gestion de contenu plutôt qu'un autre. Plus la couverture fonctionnelle est importante, plus le CMS sera adapté au projet. Cette affirmation est donc vraie.

#### Exercice p. 20 Solution n°4

Dès lors que vous appréhendez un projet Web, vous devrez, dans votre cahier des charges technique, justifier votre approche de développement. Se posera alors rapidement la question de savoir si vous allez opter pour une solution CMS existante que vous personnaliserez, ou pour un développement sur-mesure. Dans les deux cas, vous devrez trouver des arguments pour justifier vos positions.

Plusieurs arguments vous sont proposés ci-dessous. Identifiez ceux qui justifient un développement basé sur un CMS et ceux qui plaident en faveur d'un développement sur-mesure.

Pour justifier une solution basée sur un CMS	Pour justifier une solution développée sur-mesure
Réduire significativement les temps de développement.	Éviter du code inutile correspondant à des fonctionnalités non utilisées.
Proposer une solution connue qui pourra être maintenue par un tiers.	Proposer une réalisation moins vulnérable aux attaques potentielles.
Bénéficier d'une communauté de développeurs établie et de nombreuses ressources en ligne.	Garder la maîtrise sur le code développé.
Profiter de mise à jour rapide en cas de découverte d'une faille de sécurité.	Implémenter des fonctionnalités très spécifiques propres au projet.
Profiter d'une interface d'administration déjà conçue.	Proposer une interface d'administration très simplifiée.

Q Au-delà de ces arguments génériques, les spécificités de votre projet que vous aurez par ailleurs présentées dans votre cahier des charges vous aideront à compléter votre argumentaire. N'oubliez pas qu'il sera toujours pertinent de justifier vos choix, même de façon rapide.

#### Exercice p. 21 Solution n°5

Dans le cahier des charges technique, il sera important de décrire l'environnement global dans lequel évolueront les équipes techniques, mais également de détailler les trois environnements spécifiques dans lesquels les développeurs (et les autres acteurs du projet) vont évoluer. Encore faut-il bien en identifier les caractéristiques.

Complétez les phrases ci-dessous en utilisant les bons termes parmi : développement, préproduction et production.

1. L'environnement de développement est l'espace dans lequel les développeurs construisent leur code.
2. Lorsque les développements sont réalisés, ils seront placés dans l'environnement de préproduction afin que la recette fonctionnelle puisse avoir lieu.

3. Lors d'une mise à jour, on peut constater dans l'environnement de `préproduction` des effets de bord ou des régressions, qu'on pourra corriger sans affecter la version en ligne.
4. Lorsqu'on effectue la livraison finale du projet, on parle de « *mise en production* », parce qu'on place les fichiers définitifs dans l'environnement de `production`.
5. Une « *mise en préproduction* » précède la phase de recette fonctionnelle.
6. Les environnements de `préproduction` et de `production` doivent être deux environnements séparés, mais de configuration identique.
7. Avant de livrer en `préproduction`, les développeurs devraient faire des tests sur leur poste, dans leur environnement de `développement`.
8. La documentation technique de la réalisation se produit dans l'environnement de `développement`.