CONCEPTION TECHNIQUE

Spécifications Techniques Détaillées

La traduction technique des SFD : structures de données, composants, algorithmes, interfaces, au niveau où le développement peut commencer.

PROJET	. . . . .
VERSION	. . . . .
DATE	. . . . .
AUTEUR	. . . . .
RÉDIGÉ PAR	MOE / concepteur technique
DESTINATAIRES	Développeurs, testeurs techniques, mainteneurs
STATUT	Brouillon / En relecture / Validé

1 Identification et rattachement

Le lien avec le niveau fonctionnel et le suivi de version.

- Identifiant du composant, version, statut
- SFD et exigences couvertes
- Renvoi vers le DAT pour le cadre architectural

2 Rappel du besoin

Quelques phrases pour que le document se lise sans ouvrir les SFD en parallèle.

3 Découpage technique

Les composants, modules ou classes à réaliser, et leur responsabilité.

- Schéma des composants et de leurs dépendances
- Responsabilité de chacun, en une phrase
- Composants réutilisés depuis l'existant, plutôt que redéveloppés

4 Modèle physique de données

Le détail des structures de stockage, au niveau du champ.

- Tables, colonnes, types, longueurs, nullabilité, valeurs par défaut
- Clés primaires, clés étrangères, contraintes d'intégrité
- Index, avec la requête qu'ils servent à accélérer
- Volumétrie attendue par table à trois ans, et stratégie de purge

5 Description des traitements

L'algorithme, au niveau où un développeur n'a plus de décision de conception à prendre.

- Enchaînement des étapes, en pseudo-code ou en diagramme de séquence
- Transactions : périmètre, points de validation, points d'annulation
- Le traitement peut-il être relancé deux fois sans créer de doublon ni fausser un total (idempotence)
- Parallélisme, verrouillage, gestion de la concurrence

6 Interfaces techniques

Le contrat exact des échanges, opposable aux équipes voisines.

- Signatures des services exposés : méthode, paramètres, format, codes de retour
- Services consommés, avec leur comportement attendu en cas d'indisponibilité
- Formats de fichiers : structure, encodage, séparateurs, nommage, dépôt
- Contrats de message si file ou bus, avec la politique de rejeu

7 Gestion des erreurs techniques

Le comportement du composant quand son environnement se dérobe.

- Erreurs techniques prévues : indisponibilité, expiration, saturation, données illisibles
- Politique de reprise : nombre de tentatives, temporisation, abandon
- Journalisation : niveau, contenu, ce qui ne doit jamais être journalisé (secrets, données personnelles)
- Remontée vers la supervision

8 Performance et dimensionnement

Les objectifs chiffrés hérités du DAT, traduits en contraintes de conception.

- Temps de traitement cible et volumétrie de référence
- Consommation mémoire et stratégie de traitement par lots
- Points identifiés comme sensibles, et parade retenue

9 Sécurité applicative

Les dispositions au niveau du code, pas au niveau de l'infrastructure.

- Contrôle des habilitations dans le traitement
- Validation des entrées et protection contre les injections
- Gestion des secrets : mode d'accès, jamais la valeur
- Données personnelles manipulées et traitement associé

10 Tests unitaires attendus

Les cas que le développement doit couvrir, posés dès la conception.

- Cas nominaux, cas limites, cas d'erreur technique
- Jeux de données et bouchons nécessaires
- Couverture minimale attendue

11 Livraison et déploiement

Ce qui est produit et comment ça s'installe.

- Livrables : binaires, scripts, paramétrages, scripts de base de données
- Paramètres de configuration, avec valeur par environnement
- Ordre de déploiement et prérequis
- Procédure de retour arrière du composant

12 Traçabilité

Le lien entre le niveau fonctionnel et le niveau technique.

- Tableau exigence ou SFD vers composant
- Écarts assumés par rapport aux SFD, et justification

Notes de rédaction

Aide-mémoire. Cette page ne fait pas partie du document livré.

À quoi sert ce document

Les STD disent comment on construit ce que les SFD ont décrit. Elles existent parce qu'une spécification fonctionnelle, même complète, laisse ouvertes des questions qui engagent la maintenance pour des années : quelle structure de données, quel découpage en composants, quel comportement en cas d'erreur technique. Écrire ces choix avant de coder les rend discutables en revue plutôt que découvrables en production. Sur un projet interne où l'auteur des SFD code lui-même, les STD peuvent se réduire à quelques pages ; sur un développement externalisé ou repris en TMA, elles deviennent le seul lien entre l'intention et le code livré.

Quand le produire

- Sur un développement spécifique confié à une équipe distincte de celle qui a conçu.
- Quand la solution sera reprise en TMA ou par un autre prestataire.
- Sur les composants à forte complexité algorithmique ou à fort enjeu de performance.
- Quand une revue technique par les pairs est prévue avant le développement.

Erreurs les plus fréquentes

- Recopier les SFD en changeant le vocabulaire. Si les STD n'ajoutent aucune décision technique, elles ne servent qu'à alourdir la maintenance documentaire.
- Les rédiger après le développement, pour la forme. Une STD écrite après coup décrit le code au lieu de le cadrer, et n'a jamais permis d'éviter une erreur de conception.
- Ne pas dire si le traitement peut être relancé, ni comment. C'est ce qui détermine si l'exploitation pourra relancer un traitement en échec sans appeler un développeur.
- Concevoir sans volumétrie. Un traitement qui charge tout en mémoire passe la recette et tombe en production au premier vrai volume.
- Documenter les index sans dire quelle requête ils servent. Personne n'osera plus les supprimer, et ils s'accumuleront.

Modèle « STD - Spécifications Techniques Détaillées », publié par DataALC.
Version en ligne, commentée : https://www.dataalc.com/boite-a-outils/templates/std/

Adaptez ce modèle à votre contexte : supprimez les rubriques sans objet plutôt que de les laisser vides.