

Département fédéral de l'environnement, des
transports, de l'énergie et de la
communication (DETEC)
Office fédéral des routes (OFROU)
Division infrastructure routière
Filiale d'Estavayer-le-Lac

N01.12 080159
02U Bernex - Ferney

UPlaNS Bernex-Ferney

Document de Standardisa- tion Conceptuelle (DSC)

DSC-24 : Développement logiciel



OFROU

02U Bernex-Ferney

Document de Standardisation Conceptuelle

DSC-24 : Développement logiciel

Version	V1.0		
Document	U0220RA196		
Date	11.05.2020		

Table des matières		Page
1.	Introduction	2
2.	Documents de référence	2
3.	Glossaire	2
4.	Processus de développement logiciel et de génération de version	3
4.1	Exigences	3
4.2	Objectif recherché	3
4.3	Accompagnement et contrôle	3
5.	Méthodologie de conception logicielle	3
5.1	Exigences	3
5.2	Objectif recherché	4
5.3	Accompagnement et contrôle	4
6.	Codage informatique	4
6.1	Exigences	4
6.2	Objectif recherché	7
6.3	Accompagnement et contrôle	7
7.	Tests, recettes et réceptions	7
8.	Gestion des faits techniques	7
9.	Annexes	7

1. Introduction

Le présent document DSC-24 définit les exigences de développement logiciel sur l'UeLS-CH UT II sur les thématiques suivantes :

- processus de développement logiciel et de génération de version ;
- méthodologies de conception logicielle ;
- codage informatique ;
- tests, recettes et réceptions ;
- gestion des faits techniques.

Trois objectifs sont visés par l'application de ces exigences :

- uniformiser les processus de développement logiciel pour toutes les entreprises ;
- faciliter l'évolutivité du code source et sa reprise par une entreprise tierce ;
- diminuer les sources potentielles d'erreurs par l'adoption de principes vertueux.

2. Documents de référence

Se reporter au DSC-00.

3. Glossaire

Se reporter au DSC 00.

4. Processus de développement logiciel et de génération de version

4.1 Exigences

Se reporter au DSC-12.

4.2 Objectif recherché

Le cadrage strict par le DSC-12 des processus de développement et de génération de version a pour objectif de faire appliquer un cycle de vie logiciel uniformisé et vertueux à toutes les entreprises au travers d'exigences minimales :

- uniformisation de l'utilisation des différentes plateformes informatiques (le DSC-12 définit un guide d'utilisation) ;
- homogénéisation des processus de développement depuis la plateforme de développement, de génération de versions, de passage d'une version de VM d'une plateforme à l'autre (respect du process logiciel imposé par le projet) ;
- systématisation du versioning pour tous les ordinateurs ;
- au final, possibilité pour une entreprise tierce de pouvoir reprendre le processus de développement logiciel d'un ordinateur sans la péjorer.

4.3 Accompagnement et contrôle

L'accompagnement et le contrôle de l'entreprise sur les aspects liés aux processus sont assurés par l'administrateur PROC dont les fonctions sont définies au DSC-12. Celui-ci s'assure de la bonne application par chaque entreprise des concepts à appliquer.

5. Méthodologie de conception logicielle

5.1 Exigences

Il est exigé que l'entreprise utilise un outil de conception logicielle adapté et reconnu (cf. DSC-23).

L'utilisation de l'outil de conception par l'entreprise (quels modèles, dans quel but, lisibilité des graphes, lien avec les spécifications fonctionnelles et la matrice de traçabilité des exigences, etc.) est à présenter au groupe de projet au démarrage de la phase SIA-51.

NB : l'outil de conception logiciel n'est pas exigé pour :

- la rédaction des spécifications fonctionnelles de l'entreprise ;
- la génération du code informatique.

5.2 Objectif recherché

L'utilisation adéquate par l'entreprise d'un modèle reconnu de conception logicielle a un double objectif :

- garantir une conception logicielle saine de l'UeLS-CH UT II, facilement compréhensible et maintenable ;
- garantir qu'une entreprise tierce puisse reprendre facilement la conception informatique d'un ordinateur puis son codage informatique au travers de la lecture des modèles de conception utilisés (sans effectuer une rétroconception fastidieuse par analyse poussée de la documentation existante ou du code existant).

5.3 Accompagnement et contrôle

L'accompagnement et le contrôle de l'entreprise sur la méthodologie de conception logicielle sont assurés (avec le soutien du groupe de projet) :

- par l'administrateur PROC, pour ce qui concerne :
 - la validation de l'outil de conception logicielle ;
 - le contrôle de sa bonne utilisation par l'entreprise ;
 - le contrôle de l'adéquation des modèles résultants pour leur réutilisabilité par une entreprise tierce ;
- par le RL pour ce qui concerne l'adéquation aux besoins fonctionnels.

Les recommandations de l'administrateur PROC et du RL ont un caractère obligatoire.

6. Codage informatique

6.1 Exigences

Il est exigé que l'entreprise respecte les exigences de codage informatique définies sur l'UeLS-CH UT II. Leur application garantit l'usage de bonnes pratiques de réalisation informatique et participe à la qualité du code :

- lisibilité et compréhension du code ;
- maintenance du code lors de reprises pour corrections / évolutions ;
- détection aisée des problèmes dans les traitements des erreurs.

Ces exigences comportent des règles de codage informatique ainsi qu'un ensemble d'exigences définies par le groupe de projet (voir principes ci-dessous).

6.1.1 Règles de codage

Les règles de codage concernent :

- le nommage des fichiers de code source ;
- la structure et l'organisation des fichiers de code source ;
- la déclaration, l'initialisation et le nommage des variables, des instances d'objets, etc. ;
- la clarté et la systématisation de l'indentation du code ;
- la qualité des commentaires et documentation du code ;
- la gestion des erreurs logicielles ;
- la recherche de l'optimisation du code en termes de performance ;
- etc.

De nombreux guides de bonne pratique de codage informatique existent dans la littérature, les références applicables sur l'UeLS-CH UT II seront données à l'entreprise au cours de la phase SIA-51.

Quelques exemples de règles de codage assurant la maintenance et la pérennité du code :

- **l'utilisation d'adresses IP dans le code est proscrite ;**
- l'identification d'une variable doit commencer par une lettre minuscule indiquant son type ;
- toute variable doit être déclarée même pour les langages permettant des créations de variables à la volée sans déclaration formelle ;
- la déclaration des variables est toujours réalisée en début de méthode (pas à la volée dans le code) ;
- aucune variable globale ne doit être utilisée ;
- l'identification d'une classe doit être écrite en "CamelCase" en commençant avec une lettre majuscule ;
- l'identification d'une méthode doit être écrite en "camelCase" en commençant avec une lettre minuscule ;
- l'identification d'une méthode ou fonction doit contenir un verbe ;
- l'identification d'une constante doit être écrite intégralement en majuscule ;
- toute variable, instance doit être initialisée ;
- l'indentation du code est systématique pour tous les codes sources ;
- pour les langages ne nécessitant pas, dans tous les cas, des accolades après des instructions type "if" et "else" doivent obligatoirement avoir des accolades ;
- pour des instructions "switch case", définir tous les cas possibles ;
- aucun nombre magique ne doit être utilisé ;
- des règles de nommage SQL sont à respecter (noms des tables, etc.) ;

- les lignes contenant deux instructions doivent être décomposées en deux lignes. Par exemple l'instruction "if ((a=b) == 0) {" doit être décomposée comme suit :

```
a=b;  
if (a = 0) {
```

- etc.

6.1.2 Exigences définies par le groupe de projet

Les exigences complémentaires suivantes sont à appliquer :

1. Remise au MO de l'ensemble du code source réalisé.
2. Structuration des développements garantissant des traitements réactifs (optimisation du code).
3. Distinction du code technique par rapport au code applicatif. Un changement de protocole ne doit pas impacter le code applicatif.
4. Utilisation du français dans le code source (noms de variables, de fonctions, de méthodes, etc. ainsi que des commentaires). Une écriture en anglais doit faire l'objet d'une approbation par le groupe de projet et peut être refusée.
5. Présentation de commentaires pour tout élément de code afin d'en faciliter la compréhension. Leur contenu et format sont normalisés et définis au cours de la phase SIA-51.
6. Présentation de commentaires en tête de chaque fonction/méthode pour préciser le rôle, les paramètres d'entrée et de sortie.
7. Constitution d'un lexique avec toutes les variables et fonctions utilisées : à quoi sert telle ou telle variable, telle ou telle fonction.
8. Approbation du modèle conceptuel de données par le groupe de projet.
9. Utilisation de variables au format texte lorsque les échanges de données OPC-UA nécessitent des données structurées (DSC-20).
10. Utilisation d'outils de revue automatique du code source donnant un aspect qualitatif / quantitatif avec propositions d'améliorations et application régulière par l'entreprise (fréquence bi-hebdomadaire au démarrage de la phase SIA-52 puis mensuelle par la suite). Les rapports d'analyse statique sont fournis à l'auditeur du code source qui peut effectuer lui-même un audit du code source et exiger des mesures correctives.
11. Écriture du code source permettant sa reprise par n'importe quelle entreprise pour des évolutions futures.
12. Adaptation obligatoire aux exigences projet de tout code existant repris sur l'UeLS-CH UT II.

L'ensemble des règles de codage et exigences projet doit être appliqué et décrit dans un document de spécifications lors de la phase SIA-51 puis soumis à validation du groupe de projet.

6.2 Objectif recherché

L'application des exigences vise à assurer la qualité du code informatique dans le but de :

- diminuer les sources potentielles d'anomalies, réduire les durées de diagnostic et de résolution des anomalies ;
- faciliter la reprise du code source par une entreprise tierce.

6.3 Accompagnement et contrôle

Des contrôles par l'auditeur source, faisant partie du groupe de projet, sont effectués tout au long du projet. Les interfaces avec l'auditeur du code source sont définies dans le DSC-23.

7. Tests, recettes et réceptions

Se reporter au DSC-23.

8. Gestion des faits techniques

Se reporter au DSC-23.

9. Annexes

Sans objet.