

Projet GL - Description des produits du rendu final

État des documentations et de l'architecture du projet

Équipe : *gl43*

15 janvier 2026

1 Objet du document

Ce document récapitule les livrables attendus en fin de projet (documentations et bilans) et précise leur contenu visé et leur état d'avancement actuel. Il sert de support de suivi et de plan de rédaction.

2 Contexte technique et état global

Le compilateur Deca est structuré selon les trois phases classiques de compilation :

- **Étape A** : analyse lexicale/syntaxique et construction de l'AST.
- **Étape B** : vérifications contextuelles et décoration de l'AST.
- **Étape C** : génération de code assembleur IMA à partir d'un AST décoré.

2.1 État d'avancement au moment de la rédaction

- Étape A : **terminée et validée**.
- Étape B : **terminée et validée**.
- Étape C : **en cours de finalisation** (correction des bugs levés lors des tests, factorisation de code, implémentation de `instanceof` et de `cast` et enrichissement de la base des tests).

2.2 Organisation générale des documents à produire

Le tableau suivant synthétise les produits attendus et leur statut actuel.

Livrable	Finalité	Statut actuel
Manuel utilisateur	Décrire l'usage, options, erreurs, limitations	Non rédigé (plan prêt)
Documentation de conception	Expliquer architecture, choix, algorithmes	Partiel (architecture stabilisée, C à finaliser)
Documentation de validation	Expliquer stratégie, scripts, couverture, risques	Non rédigé (stratégie cependant déjà élaborée)
Documentation d'extension	Spécifier et valider l'extension choisie	Commencée, sujette à des modifications
Analyse impacts énergétiques	Mesures et discussion (code + processus)	Non rédigé (données à collecter)
Bilan gestion équipe/projet	Analyse a posteriori de l'organisation	Non rédigé (sera fait en fin)

3 Documents à rendre

3.1 Manuel utilisateur

État d'avancement. La rédaction n'a pas encore commencé. Le manuel sera rédigé après stabilisation complète de l'étape C, afin de garantir la cohérence entre les fonctionnalités décrites et le comportement réel du compilateur.

Contenu prévu. Le manuel s'adressera à un utilisateur du compilateur et comprendra notamment :

- **Périmètre supporté** : sous-ensembles du langage Deca effectivement pris en charge (hello-world, sans-objet, complet).
- **Utilisation de decac** : commandes, options, conventions (entrées/sorties, organisation des fichiers générés).
- **Messages d'erreur** : exemples représentatifs pour erreurs lexicales, syntaxiques, contextuelles, et erreurs à l'exécution (IMA), avec conditions de déclenchement.
- **Limitations connues** : comportements non implémentés, cas limites, restrictions éventuelles, et recommandations de contournement.
- **Extension** : description et mode opératoire.

Fichier attendu (dépôt Git). docs/Manuel-Utilisateur.pdf

Date limite de rendu. Lundi 19 janvier à 20h

3.2 Documentation de conception

État d'avancement. L'architecture globale du compilateur est stabilisée. L'étape A (AST) et l'étape B (décoration / vérification contextuelle) sont finalisées. L'étape C étant en cours de finalisation, certains détails de conception liés à la génération de code (cas limites, conventions finales, optimisations éventuelles) seront consolidés une fois le code stabilisé.

Contenu prévu. La documentation de conception décrira principalement :

- **Architecture** : composants principaux (A/B/C), responsabilités, dépendances (unidirectionnelles), organisation des passes.
- **Organisation interne** : répartition en packages, conventions de nommage et points d'extension.
- **Choix de conception** : décisions importantes (décoration AST, environnements, gestion registres/pile, conventions IMA), avec justifications.
- **Algorithmes et structures de données spécifiques** : uniquement les éléments non triviaux).

Fichier attendu (dépôt Git). docs/Documentation-Conception.pdf

Date limite de rendu. Vendredi 23 janvier, au moment de la soutenance

3.3 Documentation de validation

État d'avancement. Les jeux de tests et scripts sont utilisés tout au long du développement (tests par étape, non-régression, exécution CI). La documentation n'est pas encore rédigée : elle sera formalisée une fois l'étape C stabilisée, pour refléter exactement l'état final des scripts, des dossiers et des métriques de couverture.

Contenu prévu. L'objectif étant de permettre à quiconque de reproduire la validation et d'identifier les lacunes de couverture, la documentation inclura :

- **Descriptif des tests** : types de tests par étape/passe (A, B, C), organisation des répertoires et objectifs.
- **Scripts de tests** : procédures pour lancer l'ensemble des tests de manière automatisé, interprétation des sorties
- **Gestion des risques et des rendus** : suivi des risques techniques/organisationnels.
- **Couverture** : résultats Jacoco (classes couvertes, points faibles identifiés).
- **Validation complémentaire** : méthodes autres que le test (revue croisée, tests métamorphiques, cas extrêmes, etc.).

Fichier attendu (dépôt Git). docs/Documentation-Validation.pdf

Date limite de rendu. Vendredi 23 janvier, au moment de la soutenance

3.4 Documentation de l'extension

État d'avancement. Implémentations débutées, certaines finalisées. La documentation est à l'état d'ébauche et sera détaillée ultérieurement lorsque les implémentations seront terminées et validées.

Contenu prévu. La documentation d'extension (20–30 pages) inclura :

- **Spécification** : fonctionnalités et périmètre exact, hypothèses, limites.
- **Analyse bibliographique** : sources, comparaison des approches.
- **Conception / architecture** : intégration dans A/B/C , choix techniques.
- **Algorithmes** : description détaillée et justification.
- **Méthode de validation** : protocole de test, critères d'acceptation.
- **Résultats** : synthèse des validations et discussion.

Fichier attendu (dépôt Git). docs/Documentation-Extension.pdf

Date limite de rendu. Vendredi 23 janvier, au moment de la soutenance

3.5 Analyse des impacts énergétiques du projet et propositions

État d’avancement. La réflexion est en cours. Les mesures chiffrées (temps CPU, cycles IMA, etc.) seront collectées en fin de projet lorsque la chaîne de compilation et les tests seront stabilisés, afin d’obtenir des données représentatives.

Contenu prévu. Le document (4–10 pages) abordera :

- **Impact des choix de compilation** : discussion sur les choix de génération de code (coût en cycles, gestion des registres, empilements).
- **Impact du processus de validation** : stratégie de tests (sélection intelligente, scripts “éco”), compromis entre qualité et coût énergétique.
- **Extension** : prise en compte de l’impact énergétique de l’extension.

Fichier attendu (dépôt Git). docs/Impact-Energetique.pdf

Date limite de rendu. Vendredi 23 janvier, au moment de la soutenance

3.6 Bilan de gestion d’équipe et de projet

État d’avancement. Le bilan sera rédigé en fin de projet afin de permettre une analyse critique complète (organisation, planification, écarts, enseignements).

Contenu prévu. Le bilan inclura :

- **Organisation de l’équipe** : rôles, méthodes de travail (revues, pair programming), gestion Git/CI.
- **Planification vs réalisation** : incréments, priorités, ajustements, justification des écarts.
- **Analyse critique** : points d’amélioration, retours d’expérience.

Fichier attendu (dépôt Git). docs/Bilan-Projet.pdf

Date limite de rendu. Mercredi 21 janvier à 9h

4 Plan de production des documentations

Afin de rester cohérent avec l’implémentation finale, l’équipe adopte l’ordre de production suivant :

- Stabilisation de l’étape C (correction + non-régression + intégration) ;
- Rédaction du **manuel utilisateur** (lancement des tests, options, erreurs, limitations) ;
- Rédaction de la documentation de **conception** (architecture, éventuellement sous format UML) ;
- Rédaction de la documentation de **validation** (scripts + couverture Jacoco + gestion risques/rendus) ;
- Rédaction du **bilan énergie** (mesures et analyse sur base stable) ;
- Rédaction du **bilan projet** (analyse a posteriori) ;
- Documentation d’extension en parallèle selon l’avancement et la validation des spécifications.

5 Points à discuter au suivi 3

- Validation par l'enseignant de l'architecture globale et de la séparation A/B/C.
- Priorisation des tâches restantes sur l'étape C (cas limites, robustesse, intégration).
- Stratégie de rédaction : valider le plan des documentations et l'ordre de production.
- Extension : périmètre final, stratégie de validation.