

Interface souhaité

Mise en contexte

Dans le cadre du développement d'un outil pour analyser et suivre la dette technique, nous avions à implémenter une interface web afin d'y présenter les résultats d'analyse et de suivi. Au début du projet, le superviseur du projet a fourni une maquette de l'interface souhaitée, contenant des exemples de fonctionnalités souhaitées.

Technical Debt Evolution Dashboard – Canvas Mockup
You're viewing user-generated content that may be unverified or unsafe. Report

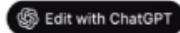


Tableau de bord — Dette Technique Git-based

Filtres & exploration
Sélectionnez le dépôt, la période et les métriques à analyser.

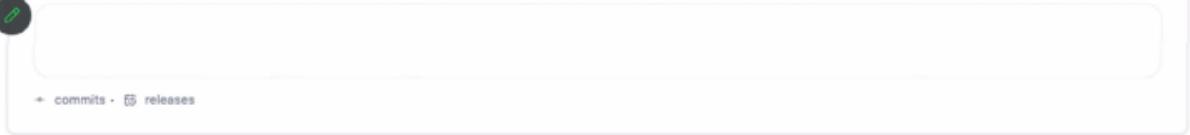
Dépôt: acme/monorepo | Période: 90 jours | Branche: main | 

Inclure TODO/FIXME Inclure complexité Inclure duplication

Total instances de dette 1,248	Fichiers à risque élevé 37	Bogues liés 112	Coût de maintenance (tendance) +18%
--	--------------------------------------	---------------------------	---

Évolution de la dette dans le temps
Somme pondérée (TODO/FIXME, complexité, duplication) par commit/release.

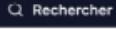




Corrélation dette ↔ bogues
Chaque point = fichier / module (x: sévérité de dette, y: # bogues, couleur: activité de commits).



Recherche
Trouver un fichier, une fonction ou un TODO.

ex: src/utils/parser.cpp ou FIXME: memory leak 

Fichiers à prioriser
Trié par sévérité estimée (dette, bogues, complexité, activité).

Fichier	Dette	Bogues	Top risques	Récemment dégradés	Stables
src/utils/parser.cpp	▲ 78	✖ 12	Complexité	Activité	
src/core/cache_manager.cpp	▲ 65	✖ 9	Complexité	Activité	

The screenshot shows a software interface for code analysis. At the top, there are two sections with file metrics:

Fichier	Lignes	Problèmes	Complexité
src/ui/components/graph.tsx	54	6	12
services/auth/tokon.go	41	7	16

Below these are two main sections:

- Recommendations**: Actions suggérées sur la base des métriques.
 - Découper src/utils/parser.cpp (fonction > 200 lignes, TODO obsolètes).
 - Complexité TODO
 - Planifier une campagne de refactor pour modules les plus actifs avant release.
 - Activité Release
 - Suivre duplication dans core/ (soupçon de copier/coller récurrent).
 - Duplication
- À faire (Backlog)**: Propositions d'extensions par les étudiants.
 - Extraction automatique des versions à intervalles réguliers.
 - Détection de duplication par hashing/AST.
 - Liaison bugs ↔ commits (Jira/GitHub Issues).
 - Heatmap par auteur / module.
 - Prédiction du risque via régression/clustering.