

Séquence 1 — Du besoin aux exigences vérifiables

Cahier d'analyse snapguard — document enseignant

Ressource R3.03 — BUT2 Informatique

September 1, 2026

Contents

1	L'enquête : de la demande au besoin	1
1.1	La demande initiale	1
1.2	Extrait de l'entretien	1
1.3	Les « 5 Pourquoi »	2
1.4	Ce que l'enquête a fait apparaître	2
2	Les exigences	2
2.1	Glossaire du domaine (le langage ubiquitaire)	2
2.2	Règles de gestion	2
2.3	Exigences fonctionnelles	3
2.4	Exigences non fonctionnelles	3
2.5	L'exercice de bascule EF ↔ ENF	4
2.6	Hors périmètre (v1)	4
3	Notes pour la conduite des trois séances	5
3.1	CM1	5
3.2	TD1	5
3.3	TP1	5

Ce que couvre cette séquence

CM1	pourquoi l'analyse ; demande vs besoin ; les 5 Pourquoi ; EF et ENF ; ISO 25010
TD1	confrontation des lectures, jeu de rôle de l'entretien, rédaction des exigences
TP1	laboratoire des liens durs, <i>spike</i> de performance, socle du projet

Le contexte, rappel

Le laboratoire de recherche *BioSeq* (18 chercheurs, 4 doctorants) traite des jeux de données génomiques sur une dizaine de postes Linux et macOS. Il n'a pas de service informatique : Karim, ingénieur d'études, gère les machines « en plus » de son travail.

Le contexte complet, le courriel de Karim et la transcription intégrale de l'entretien constituent le **dossier client**, distribué aux étudiants avant le CM1.

1 L'enquête : de la demande au besoin

1.1 La demande initiale

Karim : « Il me faudrait un script qui fasse un `tar .gz` de `/data` toutes les nuits sur le NAS. Ça devrait être vite fait, non ? »

Trois mots trahissent une solution déjà choisie : *script*, *tar.gz*, *toutes les nuits*. Aucun ne dit **quel problème** est résolu. On enquête.

1.2 Extrait de l'entretien

Analyste : Racontez-moi la dernière fois où l'absence de sauvegarde vous a posé problème.

Karim : Il y a trois semaines. Une doctorante a lancé un script de nettoyage avec un mauvais chemin. Il a écrasé six mois de résultats intermédiaires. On s'en est aperçu... quatre jours plus tard.

Analyste : Qu'auriez-vous voulu pouvoir faire à ce moment-là ?

Karim : Récupérer le dossier tel qu'il était le jeudi précédent. Juste ce dossier, pas toute la machine.

Analyste : Et si vous aviez eu vos archives *tar.gz* de chaque nuit ?

Karim : ... Il aurait fallu que je sache dans laquelle chercher, que je la décompresse en entier — on parle de 400 Gio — pour en sortir un dossier. Et de toute façon je n'ai pas la place pour trente archives de 400 Gio. C'est bien pour ça que je n'en garde que deux, en fait.

Analyste : Donc aujourd'hui, votre profondeur d'historique réelle est de deux jours ?

Karim : (silence) Deux jours. Et l'incident a été découvert au bout de quatre.

1.3 Les « 5 Pourquoi »

1. *Pourquoi un tar.gz nocturne ?* — Pour avoir une copie des données.
2. *Pourquoi une copie ?* — Pour restaurer en cas de perte.
3. *Pourquoi cela n'a-t-il pas marché la dernière fois ?* — Parce que la perte a été détectée après l'écrasement des deux seules archives conservées.
4. *Pourquoi n'en garder que deux ?* — Parce que chaque archive fait la taille complète des données.
5. *Pourquoi une archive complète pour des données qui bougent peu ?* — Parce que *tar* ne sait pas faire autrement... **et c'est la cause racine.**

Cause racine

Le besoin n'est pas « faire une copie », c'est « **pouvoir revenir à l'état de n'importe quel fichier à n'importe quelle date des N derniers jours, avec l'espace disque dont on dispose** ». La contrainte d'espace n'est pas un détail d'implémentation : c'est *elle* qui fait échouer la solution demandée.

1.4 Ce que l'enquête a fait apparaître

#	Problème métier constaté	Preuve
P1	profondeur d'historique (2 j) inférieure au délai de détection d'un incident (4 j)	incident du 10 août
P2	restaurer un seul fichier impose de décompresser 400 Gio	déclaration de Karim
P3	personne ne vérifie que la sauvegarde de la nuit a réussi	question de fin d'entretien
P4	le parc est à la fois Linux et macOS	inventaire

✗ À ne pas faire : Le piège dans lequel il ne fallait pas tomber

Répondre « d'accord » à la demande initiale aurait produit un script de 15 lignes, livré en une heure, **techniquement irréprochable et métier-ment inutile** : il aurait reproduit exactement la situation qui a causé la perte de données. Le coût de l'erreur n'aurait été visible qu'au prochain incident — c'est-à-dire, selon Boehm, au tarif « production ».

2 Les exigences

2.1 Glossaire du domaine (le langage ubiquitaire)

Terme	Définition retenue
Source	Répertoire dont on veut préserver l'état.
Dépôt	Répertoire, sur un autre support, qui héberge les instantanés.
Instantané	Image complète et navigable d'une source à une date donnée.
Politique	Règles de sauvegarde d'une source : fréquence, dépôt, rétention, exclusions.
Rétention	Durée de conservation d'un instantané avant purge.
Restauration	Action de remplacer un fichier ou un répertoire depuis un instantané.

Ce tableau n'est pas de la décoration : Karim disait « backup » pour désigner tantôt le dépôt, tantôt un instantané. Une heure de discussion a été économisée sur toute la suite du projet.

2.2 Règles de gestion

- **RG-01** — Un instantané est **immuable** : une fois terminé, aucune opération ne le modifie (seule la purge peut le supprimer entièrement).
- **RG-02** — Un fichier est réputé **inchangé** si sa **taille** et sa **date de modification** sont identiques à celles de l'instantané précédent.
- **RG-03** — Un fichier inchangé n'est pas recopié : l'instantané pointe vers l'exemplaire déjà présent dans le dépôt.
- **RG-04** — Un instantané interrompu n'est jamais présenté comme complet.
- **RG-05** — La purge ne peut jamais supprimer le dernier instantané complet d'une source.

 **Cas d'étude : Une règle de gestion est une décision, pas une évidence**

RG-02 est un compromis, et il faut le dire au client : un fichier modifié puis restauré à sa taille d'origine, avec `touch` pour remettre la date, passerait inaperçu. La détection sûre (empreinte SHA-256) coûte une lecture intégrale de 400 Gio chaque nuit. Karim a tranché : taille + date, avec une option `--verifier` pour un contrôle mensuel complet.

Faites verbaliser ce choix par les étudiants. C'est le meilleur exemple qu'ils rencontreront de l'analyse comme arbitrage *coût / risque*, et non comme recherche de la solution parfaite.

2.3 Exigences fonctionnelles

Réf.	Exigence	Vérifiable par
EF-01	créer un instantané d'une source dans un dépôt	<code>diff -r</code> avec la source ne signale aucun écart
EF-02	ne recopier que les fichiers modifiés ou nouveaux depuis l'instantané précédent	le rapport indique <i>n</i> copies pour <i>n</i> fichiers modifiés
EF-03	produire un rapport : fichiers examinés, copiés, liés, en erreur	lecture de la sortie standard
EF-04	lister les instantanés d'une source, avec date et taille	commande <code>snapguard liste</code>
EF-05	restaurer un fichier ou un répertoire depuis un instantané choisi	comparaison octet à octet
EF-06	supprimer les instantanés dépassant la rétention de la politique	comptage après exécution
EF-07	exclure les chemins correspondant aux motifs d'exclusion	aucun <code>*.tmp</code> dans le dépôt
EF-08	signaler par un code de retour non nul tout instantané incomplet	<code>echo \$?</code>

2.4 Exigences non fonctionnelles

Passage en revue systématique des huit caractéristiques **ISO/IEC 25010**. Les cases vides sont des décisions, elles aussi.

Caractéristique	Réf.	Exigence	Métrique
Performance	ENF-PERF-01	un instantané incrémental de 500 000 fichiers dont 1 % modifiés s'achève en < 30 min	chronométrage
Performance	ENF-PERF-02	≤ 64 Mio de mémoire résidente quelle que soit la taille de la source	time -v
Fiabilité	ENF-FIA-01	une interruption ne rend jamais illisible un instantané antérieur	injection de panne
Fiabilité	ENF-FIA-02	≥ 99 % d'instantanés nocturnes réussis sur un mois glissant	journal
Sécurité	ENF-SEC-01	droits du dépôt ≤ droits de la source ; dépôt en 0700	stat
Sécurité	ENF-SEC-02	aucune donnée ne transite par un service tiers non maîtrisé	revue d'architecture
Compatibilité	ENF-COMP-01	un instantané est exploitable avec ls, cp, diff, tar sans l'outil	scénario d'acceptation 3
Portabilité	ENF-PORT-01	compile et passe les tests sur Linux et macOS, sans directive conditionnelle hors POSIX.1-2008	CI sur 2 cibles
Utilisabilité	ENF-UTI-01	utilisable sans interaction depuis cron : stdout/stderr séparés, code de retour significatif	exécution en crontab
Maintenabilité	ENF-MAIN-01	logique métier testable sans accès au système de fichiers ; couverture ≥ 80 %	gcov

🔗 Cas d'étude : La caractéristique qu'on oublie toujours

ENF-COMP-01 est la plus importante du lot — et c'est celle qui n'apparaît jamais spontanément. Elle dit : *si notre outil disparaît, les données restent récupérables*. Une sauvegarde dans un format propriétaire illisible sans son logiciel est un piège classique du marché. C'est aussi ce qui disqualifie définitivement plusieurs solutions envisageables, alors qu'aucune EF ne le faisait.

2.5 L'exercice de bascule EF ↔ ENF

Prenons : « le système doit calculer une empreinte cryptographique de chaque fichier ».

- **Dans snapguard** : l'utilisateur ne demande pas d'empreintes, il demande de retrouver ses fichiers. L'empreinte n'est qu'un *moyen de garantir la fiabilité* de la détection de changement → **ENF**.
- **Dans un outil de contrôle d'intégrité** (aide, tripwire) : produire et comparer les empreintes est la finalité du produit → **EF**.

Et l'inverse : « la sauvegarde doit s'exécuter chaque nuit à 2 h ». ENF de disponibilité ? Non — si l'outil embarque son propre ordonnanceur, la planification devient une fonctionnalité (EF). Ici, on a délibérément choisi de la déléguer à cron : elle **sort du périmètre**, ce qui est une troisième réponse possible, et souvent la meilleure.

2.6 Hors périmètre (v1)

Écrit noir sur blanc pour prévenir la dérive du périmètre — le mal qui a tué le VCF du FBI : pas de sauvegarde distante (SSH) ; pas de chiffrement ; pas d'interface graphique ; pas de gestion des liens symboliques, fichiers spéciaux, ACL ni attributs étendus ; pas de déduplication entre sources différentes.

3 Notes pour la conduite des trois séances

3.1 CM1

Le fil de la séance est celui du polycopié : le coût de Boehm, les quatre échecs, puis la distinction demande / besoin illustrée par les 5 Pourquoi de Karim. Terminez par les EF/ENF et le balayage ISO 25010 — c'est ce dont le TD1 a besoin, et rien de plus.

✓ À faire : Le moment à ne pas manquer

La classification du piège (« le système doit créer un lien dur plutôt qu'une copie »). Laissez la classe se diviser entre EF et ENF, puis tranchez : **ni l'une ni l'autre**, c'est une solution déguisée en exigence. Les étudiants n'ont pas encore fait le TP1 et ne savent pas ce qu'est un lien dur : c'est sans importance, ils jugent la *forme* de l'énoncé.

3.2 TD1

Déroulé en quatre temps : confrontation des surlignages (15 min), jeu de rôle (20 min), rédaction des exigences et balayage ISO 25010 (25 min), classification EF/ENF (15 min).

L'aide-mémoire pour jouer Karim et les huit énoncés à projeter sont dans le document *Matériel de séance*.

📖 Cas d'étude : Ce que le jeu de rôle produit vraiment

Karim répond **strictement** à ce qu'on lui demande. Une question fermée obtient « oui » ; une question suggestive obtient un acquiescement poli, et le groupe repart avec une fausse validation. C'est cette dernière situation qui marque le plus : ils comprennent qu'on peut sortir d'un entretien satisfait et complètement à côté.

3.3 TP1

Le TP1 n'est pas une séance de programmation : c'est un *spike*, une investigation technique destinée à réduire l'incertitude sur une exigence avant de s'engager. Le code produit au laboratoire est jetable ; ce qui reste, c'est la décision informée.

30 min	laboratoire d'observation : inodes, compteur de liens, les deux limites du lien dur
15 min	confrontation des ENF chiffrées du TD1 à une mesure réelle
30 min	socle du projet : dépôt, Makefile, parcours récursif

✓ À faire : La vérification de cinq minutes qui vaut pour tout le semestre

Passez dans les rangs et faites afficher `git config user.email` sur chaque poste. La note individuelle repose entièrement sur le journal Git : un dépôt à un seul auteur découvert au moment de noter est irrattrapable.

Vérifiez aussi que vous avez bien été ajouté aux dépôts (rôle *Maintainer* ou *Reporter*) : c'est demandé dans le travail préparatoire, tous ne l'auront pas fait.

✗ À ne pas faire : Ne donnez pas la réponse au laboratoire

La tentation est grande d'expliquer les liens durs avant de lancer l'exercice. Ne le faites pas : la question **Q2.4** demande d'expliquer le mécanisme à Karim *sans employer les mots* « inode » et « lien dur ». Elle n'a de sens que si la découverte a été la leur.