

Scripting Shell

Module 01 – Analyse



1

Scripting Shell



Objectifs

- Découvrir l'ingénierie logicielle
- Mettre en place un cahier des charges
- Planifier un algorithme et son pseudo-code

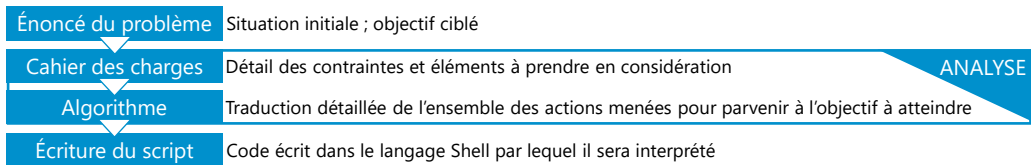


2

Introduction



- Les scripts Shell sont des **fichiers texte** contenant un ensemble de **commandes Shell**.
- Avant de se lancer dans l'écriture d'un script, un travail de **réflexion** s'impose. Cette phase d'analyse intègre la rédaction de cahier des charges et d'algorithme.



- Viendra ensuite la phase d'écriture qui vous permettra de pratiquer l'éditeur de texte **vim**.



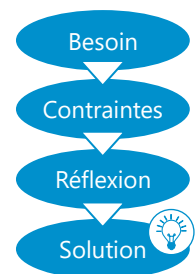
Énoncé du problème et cahier des charges



5

Énoncé du problème et cahier des charges

- La création d'un script Shell résulte de la réponse à un problème ou à un besoin donné.
- Les scripts Shell peuvent être utilisés afin de répondre à des besoins :
 - D'administrations (pour le SI)
 - De l'utilisateur
- Après analyse du besoin et des contraintes, un travail de réflexion permettra l'élaboration d'une solution.



6

Identifier le besoin → Cibler, détailler et formaliser le besoin

- Le porteur du besoin ne se préoccupe pas des contraintes techniques liées à la réalisation du script. Il décrit sa problématique et son besoin.
- Le porteur du besoin détaille et formalise ses attentes.



Exemple de description du besoin

« Nous avons besoin d'un outil qui affiche les derniers échecs de connexion utilisateur. »



Contraintes et cahier des charges → Établir un cadre au projet

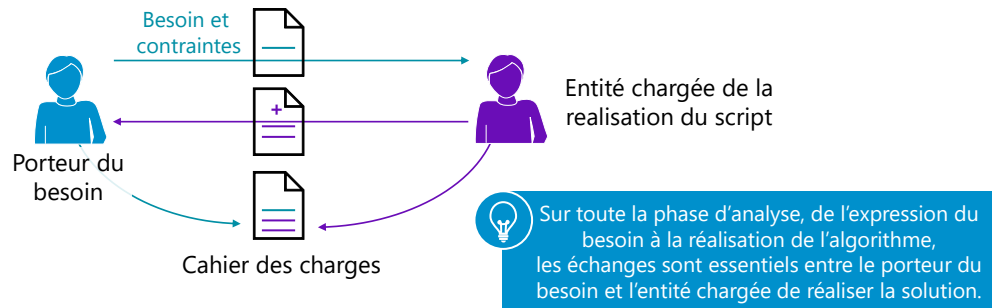
- Une fois le besoin bien compris, il reste à établir un cadre au projet.
- Ce cadre reprendra l'identification du besoin et la complètera par la liste de l'ensemble des contraintes s'appliquant à la solution.



Entité chargée de la réalisation du script



Contraintes et cahier des charges → Établir un cadre au projet



Contraintes et cahier des charges → Établir un cadre au projet

Exemple de cahier des charges de l'outil souhaité

- Il sera utilisé sur un serveur sous Debian 10
- Il n'aura pas de dépendances
- Il sera écrit en Bash
- Le nombre souhaité d'échec(s) à afficher sera défini par l'utilisateur au lancement du script
- À défaut d'indication de valeur de nombre d'échecs, une valeur par défaut sera considérée
- Le lancement du script pourra être initié par l'utilisateur ou automatisé



Réflexion ➔ Déterminer comment parvenir au résultat attendu

S'ensuit un travail d'analyse lors duquel :

- Sera étudié dans le détail le cahier des charges
- On réfléchira aux traitements à effectuer pour parvenir au résultat attendu et à l'ordonnancement des instructions



Algorithme



Pour poursuivre l'analyse, on réalisera un algorithme du script ou programme envisagé.

Définition d'un algorithme

Pour qu'un ordinateur réalise une tâche, qu'elle soit basique ou très complexe, il est nécessaire de lui indiquer dans les moindres détails toutes les actions à réaliser.

Un algorithme, c'est une description complète et détaillée des instructions ordonnancées dont l'exécution conduit à un résultat donné.

Éléments utilisés pour la réalisation d'algorithmes

- Instructions
- Variables
- Tests
- Boucles



Un algorithme peut être représenté sous forme de texte. On parle alors de **pseudo code**. L'utilisation de **schéma** ou **organigramme** est aussi adaptée à sa représentation.

Utilisation d'algorithme dans la vie de tous les jours



Les algorithmes sont aussi présents dans nos contextes professionnels.

Ils représentent une étape essentielle au processus de création d'un script Shell.



Pseudo code

Le pseudo code est utilisé pour la representation textuelle d'un algorithme

Lancement du script

log ← /var/log/secure
nbdefault ← 42

Si (nombre d'argument passé au script = 1)
alors **nb** ← argument
Finsi

Si ((**nb** est nul) OU (**nb** n'est pas un entier))
alors la **nb** ← **nbdefault**
Finsi

Afficher la liste des nb derniers échecs de connexion de **log**
Journaliser « exécution de logonfails.sh »

Fin d'exécution du script

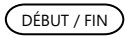

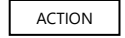


Exemple de pseudo
code de l'outil



Algorithme

La représentation sous forme d'organigrammes est basée sur l'utilisation d'un ensemble de symboles.

Les principaux symboles d'un organigramme

Symbole	Signification
 DÉBUT / FIN	L'ovale matérialise les étapes du début et de fin d'un programme.
	La flèche indique le sens des flux.
 ACTION	Le rectangle représente la réalisation d'une action ou d'un processus.
 DÉCISION	Le losange est utilisé pour représenter une décision à prendre ; ou la vérification d'une condition. Il est parfois utile d'utiliser des formes plus complexes quand on a plus de 3 décisions à prendre.
 ENTRÉE / SORTIE	Entrée / Sortie



Exemple d'algorithme de l'outil

