



DICKO INDUSTRIES

Dicko Soka

De l'idée au produit :
le parcours en seize étapes

Ils vous vendent le cheval.

Nous vous apprenons à tenir la bride.

Dicko Soka

De l'idée au produit :
le parcours en seize étapes

Mohamed Nassirou **SOUMANOU**

Fondateur de **Dicko Industries**

ÉDITÉ PAR DICKO INDUSTRIES
PREMIÈRE ÉDITION, AOÛT 2026
DÉCRIT LE HARNAIS DICKO SOKA 3.86.0
COMPILÉ LE 18 AOÛT 2026

Je ne sais pas écrire une ligne de code

Il faut le dire avant tout le reste, parce que c'est la seule chose qui donne du poids à ce livre : je ne suis pas développeur. Je n'ai ni diplôme ni carrière dans l'informatique. Je ne sais pas écrire une ligne de code.

Je suis diplômé en Management de Terminaux Portuaire auprès de la Lloyds Maritime Academy de Londres en Angleterre et douze ans de métier dans les opérations, la logistique et les terminaux portuaires en Afrique de l'Ouest. Ce que je sais faire, c'est monter une opération qui n'existe pas encore, avec une équipe qui ne la connaît pas encore, dans un délai qui ne laisse pas de marge.

Et depuis l'université, j'ai développé une passion pour l'ordinateur, grâce notamment à AIESEC et au Campus Numérique Francophone de l'UL (là où j'ai découvert les logiciels libres et le système d'exploitation Ubuntu/Linux).

Pas une compétence : une **passion**. La distinction compte, parce qu'elle décrit exactement la position de la plupart des gens qui liront ces pages : curieux de la machine, sans jamais avoir eu le temps ni la raison d'en apprendre le langage.

Ce livre est donc écrit depuis l'autre côté du mur. Pas par quelqu'un qui vulgarise ce qu'il maîtrise depuis toujours, mais par quelqu'un qui a appris récemment, en pratiquant, et qui se souvient encore précisément de ce qui était incompréhensible la veille.

Je n'ai pas appris à coder. J'ai appris à tenir la bride.

Ce que douze ans d'opérations m'ont appris

On m'a confié plusieurs fois une chose qui n'existait pas.

En janvier 2026, un port sec à ouvrir près de Freetown en vingt-cinq jours : à mon arrivée, notre partenaire dégageait encore le terrain. En 2022, une opération d'import, stockage et exportation du coton à Adétikopé : la plateforme n'en avait jamais manutentionné, moi non plus, et le seul document qu'on m'a remis était la spécification d'empilage de l'assureur. Avant cela, j'ai eu à monter les opérations du site de dépotage des conteneurs ainsi que les opérations des entrepôts, après avoir construit la fonction Process Excellence sur un terminal à conteneurs de grande envergure à Lomé, au Togo.

À chaque fois, la même méthode : écouter avant de décider, définir le périmètre avant d'agir, verrouiller les règles non négociables *avant* la première opération, puis corriger par cycles.

En découvrant le harnais de pilotage de l'IA, je n'ai donc pas découvert une idée nouvelle : **j'ai reconnu la mienne, appliquée à un autre métier.** Un harnais logiciel est une procédure d'exploitation, avec son ordre des étapes, ses règles verrouillées d'avance et ses points de contrôle. C'est pour cela que j'ai pu l'étendre.

Novembre 2022

En novembre 2022, l'intelligence artificielle conversationnelle est mise à la disposition du grand public. J'ai été captivé immédiatement, et je ne me suis plus arrêté depuis : en autodidacte, et surtout en pratiquant, la seule manière d'apprendre un outil dont la documentation vieillit plus vite qu'on ne la lit.

La première fois que j'ai utilisé Claude, j'ai trouvé autre chose. Étant Manager, ce qui m'intéressait n'était pas la performance technique : c'était la capacité à raisonner sur un problème mal posé, à structurer ce qui arrive en désordre, à me contredire quand j'avais tort. Un Manager passe ses journées là-dedans. J'ai donc approfondi cet outil-là plutôt qu'un autre, et je n'ai pas cessé depuis.

Claude m'offrait la possibilité d'approfondir certains sujets, de faire du brainstorming et surtout de produire des livrables en word ou Excel de qualité professionnel. Mais quelque chose manquait, et je ne savais pas le nommer. Pour aller plus loin, surtout avec le code, il devenait évident que je n'avais pas les compétences qu'il fallait. J'ai essayé de produire une application métier (dans le domaine de la logistique) avec Claude Code. Cela m'a consommé beaucoup de temps et d'énergie sans que l'application ne prenne réellement forme. Mes abonnements à Claude Pro puis Max, n'étaient donc pas rentabilisés comme je le souhaitais.

Ce n'était pas un problème d'intelligence de la machine. C'était un problème d'**ordre et de méthode**, et c'est exactement mon métier, sauf que je ne l'avais pas encore vu.

Je savais ce que je voulais construire. Mais je n'avais pas l'outil qui me permettait de faire travailler la machine dans l'ordre qu'il faut.

Juillet 2026 : le mot « harnais »

Début juillet 2026, je tombe sur une vidéo YouTube. Un jeune développeur français, **Flavien Chervet**, y est interviewé, et il emploie un mot que je n'avais pas entendu dans ce sens : le ***harnais***.

Son constat était celui que je n'arrivais pas à formuler : le harnais est ce qui manquait aux non-techniciens pour obtenir de véritables résultats de l'IA. Pas un modèle plus puissant, pas une meilleure formulation des demandes, mais une couche au-dessus, qui sait quoi faire et dans quel ordre. Il expliquait en avoir mis un en place, et que ses étudiants non-techniciens menaient avec lui des projets de bout en bout.

Il annonçait qu'il l'ouvrirait plus tard en **open source** (logiciel dont le code est mis à disposition du public), et j'ai attendu cette ouverture.

Quand il l'a ouvert, je l'ai pris, je l'ai étudié, et j'ai commencé à l'adapter aux réalités de l'Afrique de l'Ouest en général et de la zone UEMOA en particulier.

ATTRIBUTION

Dicko Soka descend d'un harnais ouvert en open source par Flavien Chervet. L'idée du harnais, et la première mise en œuvre dont nous sommes partis, sont les siennes. Ce livre décrit ce que nous en avons fait ; il ne prétend pas à l'invention de ce dont il hérite.

Ce que nous y avons ajouté

Un harnais conçu ailleurs suppose un marché qui n'est pas le nôtre, et la première divergence est le paiement.

Là où le harnais d'origine parlait d'intégrer **Stripe**, j'ai ajouté **FedaPay**, qui encaisse par mobile money. Ce n'est pas une préférence technique : c'est ainsi que nos clients paient. Un produit ouest-africain qui n'accepte que la carte refuse la majorité de son marché.

Puis d'autres commandes, chacune née d'un manque rencontré (la liste ci-dessous n'est pas exhaustive de nos commandes propres mais elle est à titre indicatif) :

/pain-point-finder	Chercher les points de douleur documentés d'un marché, au lieu de partir d'une intuition.
---------------------------	---

/bmc	Schématiser le modèle économique, parce qu'une idée sans modèle est un passe-temps.
-------------	---

/generate • /montage	Produire et assembler les visuels, parce qu'un bon produit invisible ne se vend pas.
------------------------------------	--

Et d'autres encore, plus une réécriture continue de ce qui existait. Aujourd'hui, **Dicko Soka** est à environ **60 %** notre travail propre.

LA PREUVE

Dicko Soka m'a permis de concevoir **TransiFa Docs** sans aucun développeur. Le service prépare les déclarations en douane au Togo, avec des professionnels qui le paient au dossier.

Pourquoi je m'y investis

J'ai perçu la portée stratégique de l'intelligence artificielle, et c'est la raison de cet investissement, et surtout pour partager mes expériences et augmenter le nombre de jeunes Africains qui se forment dans ce domaine d'avenir.

Nous pouvons réaliser énormément avec cette technologie. Elle peut nous permettre de jeter des ponts dans certains secteurs, et d'emprunter des raccourcis pour réduire notre écart technologique avec les autres nations, en tirant profit notamment des modèles ouverts, DeepSeek, Qwen, GLM et les suivants.

Mais je reste lucide. Personne au monde ne mettra à notre disposition sa technologie la plus avancée. Il nous revient, en tant qu'Africains, de faire le travail nécessaire pour développer nos propres technologies, en puisant dans notre riche tradition et notre culture, dont le paradigme demeure assez large pour concevoir des technologies que personne d'autre au monde ne peut développer.

C'est une conviction personnelle. Ce livre en est la première pierre : ce que j'ai appris une fois ne doit plus être réappris par le suivant.

Mohamed Nassirou SOUMANOU

Lomé, août 2026



Soka, c'est la bride. En éwé, *só* veut dire cheval et *ka* veut dire corde.

Dicko Industries

Trois produits, une même conviction : l'outil doit porter le savoir-faire, pas seulement l'exécution. Ce qu'un professionnel apprend à ses dépens doit devenir une règle que le produit applique de lui-même, qu'il s'agisse d'un harnais logiciel, d'une déclaration en douane ou d'une clé à choc.

Dicko Soka

CE LIVRE

Le harnais, et la méthode qu'il porte : de l'idée au premier client qui paie, en seize étapes ordonnées, sans écrire une ligne de code. Transmis aussi en séances de formation à Lomé et en ligne, autour d'une communauté de bâtisseurs. <https://dickoindustries.com/soka>

TransiFa Docs

EN SERVICE

La déclaration en douane juste du premier coup, au Togo : classement SH assisté sur le tarif togolais, le droit de douane, la valeur CAF prête, la minute et le fichier XML à télécharger et importer dans SYDONIA World. Mille francs CFA le dossier, par mobile money. **Conçu avec Dicko Soka, sans aucun développeur.** <https://test.transifa.com>

Dicko Tools

BOUTIQUE EN LIGNE EN CONSTRUCTION

L'outillage automobile professionnel à Lomé et dans la CEDEAO : crics, extracteurs, outils moteur, diagnostic multimarque. Livraison Lomé en vingt-quatre heures, paiement mobile money, conseil par WhatsApp. <https://dickotools.com>

Ce livre ne traite que du premier. Le deuxième en est né, et c'est la meilleure preuve que la méthode tient : un produit vendu, conçu sans développeur, par l'auteur de ces pages.

- Préface de l'auteur
- Ce livre, et à qui il parle
- 01 Le principe
- 02 Le parcours en entier
- 03 Si votre projet n'est pas numérique
- 04 Avant de construire
- 05 Ce qui tourne sous le capot
- 06 Le démarrage du projet
- 07 Les outils du quotidien
- 08 Ajouter une capacité
- 09 Mettre en ligne, en trois temps
- 10 Trouver des clients
- 11 Bien décrire ce qu'on veut
- 12 Travailler avec la machine
- Par où commencer
- Pour finir, et cette édition

Une seule chose à retenir si vous ne retenez rien d'autre : quand vous ne savez pas quoi faire, tapez `/next`. Le harnais regarde où vous en êtes et nomme l'étape suivante, avec sa raison.

Ce livre, et à qui il parle

Vous avez une idée. Peut-être une gêne quotidienne dans votre métier, peut-être un service que personne n'offre autour de vous. Ce qui vous manque n'est pas l'idée : c'est le chemin entre elle et un produit que des gens utilisent et paient.

Ce livre décrit ce chemin. Il ne vous apprend pas à programmer, et c'est délibéré : le harnais Dicko Soka existe précisément pour que vous n'ayez pas à le faire. Il vous apprend à **quoi sert chaque étape, dans quel ordre, et pourquoi cet ordre-là.**

Et si votre projet n'est pas numérique, ne refermez pas ce livre. La moitié du parcours ne produit pas une ligne de code. Un atelier, un commerce, une auto-école se conduisent par les mêmes étapes qu'une application, parce que ce que le harnais structure n'est pas la construction : c'est la réflexion. Le chapitre 3 est écrit pour vous.

Ce que ce livre n'est pas

Ce n'est pas un manuel qui vous dit où cliquer. Les écrans changent, les manuels avec captures vieillissent en quelques semaines, et un lecteur qui suit une capture périmée conclut que l'outil est cassé alors que c'est le livre qui l'est.

Ce livre est **conceptuel**. Il explique le raisonnement derrière chaque étape, celui qui reste vrai quand un bouton se déplace. Quand vous saurez POURQUOI on isole une base de données avant de tester, la commande exacte sera la partie facile, et l'outil vous la donnera lui-même, puisqu'il est fait pour ça.

Le chiffre qui explique la méthode de ce livre

Ce livre décrit la version **3.86.0** du harnais.

64

commandes que vous tapez

41

rouages internes qui travaillent sans qu'on les appelle

105

en tout

16

étapes au parcours qu'il vous fait suivre, décrites au chapitre 2

INVENTAIRE LU DANS LE HARNAIS, VERSION 3.86.0

Ces nombres ne sont pas écrits à la main dans ce livre. Ils sont **lus dans le harnais au moment où le livre est fabriqué**, et c'est une leçon payée. Un livre imprimé avec un chiffre recopié serait faux avant même d'être téléchargé, et cette fois chez ses lecteurs, sans rattrapage possible.

Alors chaque fait de ce livre (la liste des étapes, leur ordre, les noms des commandes) est relu dans le produit à chaque fabrication. Si vous lisez un chiffre ici, il était vrai pour la version nommée en page de titre. **Vérifiez toujours cette version**, en page de titre et sur la dernière page du livre : c'est la date de péremption de ce livre.

Comment le lire

Les douze (12) chapitres suivent le parcours dans son ordre réel, de l'idée non validée jusqu'aux premiers clients. Vous pouvez les lire d'affilée, ou ouvrir directement celui de l'étape où vous êtes.

Une seule chose à retenir si vous ne retenez rien d'autre : quand vous ne savez pas quoi faire, le harnais répond à cette question lui-même. La commande `/next` regarde où vous en êtes et nomme l'étape suivante avec sa raison. C'est la seule commande qu'il faut mémoriser.

Enfin, si vous préférez ne pas commencer seul : la page **Par où commencer**, en fin d'ouvrage, dit comment nous joindre. Vous pouvez y aller tout de suite.

Le principe

Ce que vous décrivez, la machine le construit

Vous écrivez ce que vous voulez, en français, dans une conversation. La machine écrit le code, crée la base de données, met le site en ligne. Vous ne relisez pas le code : vous regardez le résultat et vous dites ce qui ne va pas.

C'est un renversement, et il mérite d'être nommé : **la compétence rare n'est plus d'écrire le code, c'est de savoir ce qu'on veut.** Décrire précisément un besoin, reconnaître un résultat correct, arbitrer entre deux options : c'est ce que vous apportez, et aucun outil ne le fait à votre place.

Pourquoi un harnais, et pas seulement un assistant

Un assistant seul répond à ce qu'on lui demande. Il le fait bien, et c'est exactement le problème : si vous ne savez pas qu'il faut isoler une base de données avant de tester, il ne le fera pas, et rien ne vous dira que vous venez de tester sur les données réelles de vos clients.

Le harnais est la couche qui **sait ce qu'il faut faire même quand vous ne le demandez pas.** Il connaît l'ordre des étapes, les pièges du marché où vous vendez, les vérifications obligatoires avant une mise

en ligne. Ses 105 commandes ne sont pas 105 fonctionnalités : ce sont 105 décisions déjà prises, testées, et qui n'ont plus à être reprises à chaque projet.

105 commandes, ce ne sont pas 105 fonctionnalités : ce sont 105 décisions déjà prises.

Trois règles qu'il ne négocie pas

Ces règles ne sont pas des préférences, ce sont des lignes rouges. Aucun réglage d'autonomie ne les lève.

01 Aucun mouvement d'argent réel sans votre accord explicite.

Pas de paiement déclenché, pas de bascule d'un système de paiement en mode réel, sans que vous ayez dit oui pour cette opération précise.

02 Aucune suppression de ressource sans votre accord explicite.

Supprimer une base, un projet, un dépôt : jamais en votre absence, jamais par déduction.

03 Aucune écriture dans la base de production sans votre accord explicite.

Et ce n'est pas une politesse : par défaut, les instructions envoyées à votre base partent en lecture seule, et c'est la base elle-même qui refuse d'écrire. Une garde qu'on peut contourner par une formulation habile n'est pas une garde.

LES TROIS LIGNES ROUGES, NON NÉGOCIABLES

Ce que ça change pour vous

Vous ne devenez pas développeur. Vous devenez quelqu'un qui livre. La différence entre une idée et un produit n'est plus une compétence technique à acquérir pendant deux ans : c'est un parcours à suivre, et le chapitre suivant le décrit en entier.

À RETENIR

Un assistant fait ce qu'on lui demande. Un harnais fait aussi ce qu'on a oublié de lui demander, et refuse trois choses, quoi qu'on lui dise : dépenser, détruire, écrire en production.

Le parcours en entier

C'est le chapitre le plus important du livre. Tous les autres décrivent une étape ; celui-ci montre pourquoi elles s'enchaînent dans cet ordre.

La carte

Le parcours n'est pas une file unique. Deux pistes avancent en parallèle : la piste **produit** construit la chose, la piste **marché** trouve les gens qui la paieront. Les étapes **communes** servent les deux.

- | | | |
|---|---|---------|
| ○ 01 | /pain-point-finder Recherche de douleurs | MARCHÉ |
| partir de douleurs documentées et sourcées, pas d'une intuition | | |
| ● 02 | /ideate Idéation critique | COMMUNE |
| valider le problème, la cible et le go/no-go avant tout code | | |
| ○ 03 | /bmc Business Model Canvas | COMMUNE |
| poser le modèle économique sur une page | | |
| ● 04 | /spec Cahier des charges | COMMUNE |
| le cahier des charges détaillé que /bootstrap consomme | | |
| ● 05 | /bootstrap Échafaudage + déploiement | PRODUIT |
| créer le projet, le déployer en ligne et valider l'infrastructure | | |
| ● 06 | /design Identité + landing | COMMUNE |
| donner une vraie identité et une landing qui convertit, sortir du template | | |
| ● 07 | /db-isolate Base de dev isolée | PRODUIT |
| travailler sur une branche de base séparée, jamais sur la prod | | |
| ● 08 | /add-staging Environnement de test | PRODUIT |
| une URL de test stable pour éprouver le produit avant tout vrai utilisateur | | |
| ● 09 | /security Audit de sécurité | PRODUIT |
| auditer avant d'ouvrir au public (verdict PASS requis pour la prod) | | |
| ○ 10 | /billing Revenu récurrent | PRODUIT |
| faire rentrer les renouvellements sans les relancer à la main | | |
| ● 11 | /gtm Plan d'acquisition | MARCHÉ |
| comment trouver et convertir les 100 premiers clients, chiffré | | |
| ○ 12 | /pitch Pitch | MARCHÉ |
| un pitch prêt pour un client, un partenaire ou un financeur | | |

○ 13	/brand-kit Kit de marque	MARCHÉ
○ 14	/promote Exécution marketing	MARCHÉ
○ 15	/metrics Mesure	MARCHÉ
● 16	/launch Mise en ligne publique	COMMUNE

FIGURE 1 : LES 16 ÉTAPES, LEUR PISTE ET LEUR RAISON

Deux pistes, pas une file d'attente

Pourquoi cette séparation existe : une session consacrée au marketing à qui on répond « va construire un environnement de test » reçoit un conseil qui n'est pas faux, mais **hors sujet**, et une boussole hors sujet se fait ignorer, puis désinstaller. Le harnais regarde donc sur quelle piste vous travaillez avant de recommander quoi que ce soit.

Requis, optionnel, et pourquoi la distinction compte

Certaines étapes sont **requis**es : sans elles, ce qui suit ne tient pas. D'autres sont optionnelles selon votre projet : un site vitrine sans comptes n'a pas besoin de tout ce qu'une application avec abonnements exige.

Le harnais ne vous bloque jamais sur une étape optionnelle. Il vous dit ce qu'elle apporterait, et vous décidez. Sur une étape requise, il insiste, et sur l'audit de sécurité avant mise en ligne, il refuse de passer outre.

Vous n'avez pas à mémoriser cette carte. Le harnais la lit pour vous.

La commande à retenir

Tapez `/next` : le harnais lit où vous en êtes, non pas ce que vous avez déclaré mais **ce qui existe réellement** sur votre disque et en ligne, puis nomme l'étape suivante et sa raison, en une ligne.

Si vous préférez qu'il enchaîne tout seul aussi loin qu'il le peut, `/pilot` fait le vol complet, s'arrête là où une décision humaine est nécessaire, et vous dit pourquoi il s'est arrêté.

À RETENIR

Seize étapes, deux pistes qui avancent ensemble. Les requises tiennent l'édifice, les optionnelles dépendent de votre projet, et `/next` vous dit toujours laquelle vient maintenant.

Si votre projet n'est pas numérique

Ce chapitre existe parce que les deux précédents ont pu vous laisser croire que ce livre parle de logiciel. Il parle de méthode. La différence décide de qui peut s'en servir.

La moitié du parcours ne produit pas une ligne de code

Reprenez la carte du chapitre précédent et regardez ce que chaque étape *fabrique*. Huit (8) des seize (16) étapes ne produisent aucun code : elles produisent des documents, des chiffres et des décisions.

01	/pain-point-finder	Un dossier de douleurs sourcées	Ce que les gens reprochent vraiment à ce qui existe, avec au moins deux sources citées et datées.
02	/ideate	Un verdict argumenté	Go, no-go ou pivot. Le problème est-il réel ? Qui exactement souffre ? Que font ces gens aujourd'hui, et pourquoi feraient-ils autrement ?
03	/bmc	Un modèle économique sur une page	Les neuf (9) blocs, dans leur ordre de dépendance, et un compte de résultat mensuel dans votre monnaie.
04	/spec	La description de votre offre	Ce que vous proposez, à quel prix, et surtout ce que vous ne proposez pas encore.
11	/gtm	Un plan d'acquisition chiffré	Qui signe en premier, par quel canal, à quel coût, avec un plan daté pour les cent (100) premiers clients.
12	/pitch	Un argumentaire	Pour un client qui hésite, un partenaire, un financeur.
13	/brand-kit	Une identité	Le nom, le logo, les couleurs, l'étiquette, l'emballage.
14	/promote	Le matériel de campagne	Les messages de prospection, le calendrier de contenu, les briefs de campagne avec ciblage et budget.

FIGURE 2 : LES HUIT ÉTAPES QUI NE TOUCHENT PAS AU CODE, ET CE QU'ELLES PRODUISENT

Un commerce, un atelier, une exploitation agricole, un service de transport, une auto-école: tous passent par ces huit étapes. Aucun n'a besoin des huit autres.

Ce que le harnais apporte vraiment

Ce n'est pas la construction. C'est **une discipline de réflexion qui refuse de vous laisser sauter les questions qui coûtent cher.**

Un cahier ne vous demande pas vos sources. Un tableur ne vous contredit pas. Un ami à qui vous exposez votre idée vous encourage, parce qu'il vous aime bien. Le harnais fait l'inverse : il exige des sources datées, il nomme la personne exacte qui paiera, il vous demande à quel prix le marché achète *aujourd'hui* et si vous l'avez vérifié ou supposé. Puis il énonce un verdict que vous n'aviez pas envie d'entendre.

Ce qui est structuré n'est pas la construction.
C'est la réflexion.

Un exemple, du début à la fin

Prenons un projet qui ne touche à aucun écran : **une savonnerie artisanale** à Lomé. Voici ce que le parcours donne.

/pain-point-finder	Ce qu'on reproche aux savons vendus sur le marché : la peau qui tire, le parfum qui ne tient pas, le prix qui monte avec l'importation.
/ideate	La contestation : combien de savonneries artisanales existent déjà à Lomé ? Qu'est-ce qui vous distingue vraiment ? À quel prix ce marché achète-t-il, et l'avez-vous vérifié ou supposé ?
/bmc	Le coût du beurre de karité et de la potasse, le prix de vente au détail et en gros, la marge par savon, le seuil de rentabilité mensuel en francs CFA.
/spec	La gamme de départ : trois savons, pas douze. Et la liste de ce qui attendra la deuxième année.
/gtm	Qui achète en premier : les boutiques de quartier, les salons de coiffure, les revendeuses de marché. Le coût et l'effort de chaque canal, et lequel commencer.
/pitch	L'argumentaire d'une minute pour la gérante d'une boutique qui hésite à prendre vos savons en dépôt.
/brand-kit	Le nom, l'étiquette, l'emballage, la façon dont le produit se reconnaît sur une étagère encombrée.
/promote	Le calendrier de publications, les messages de prospection, et la discipline qui évite de passer pour un importun.

FIGURE 3 : LE PARCOURS D'UN PROJET SANS UNE LIGNE DE CODE

Au bout de ce chemin : pas de site, pas d'application, pas une ligne de code. Un business documenté, chiffré et **contesté avant la première dépense**. C'est exactement ce que la plupart des projets ne font jamais, et c'est pourquoi la plupart des projets meurent de choses qu'on pouvait savoir d'avance.

Si un jour cette savonnerie veut vendre en ligne, les huit autres étapes l'attendent. Mais elles attendent : elles ne conditionnent rien.

Ce que le harnais ne fera pas

Il ne visite pas votre local. Il ne goûte pas votre produit. Il ne négocie pas avec votre fournisseur et ne remplace pas une matinée passée au marché à regarder ce que les gens achètent vraiment. Il structure votre réflexion ; le terrain reste le vôtre, et il reste irremplaçable.

Une limite honnête, aussi : sur un marché très local, ses sources sont plus minces que sur un marché documenté. Quand la preuve est mince, il le dit au lieu de combler le trou, et c'est à vous d'aller chercher le reste.

À RETENIR

Huit étapes sur seize ne touchent pas au code. Un savon et une application passent par les mêmes, parce que ce qui se structure n'est pas ce qu'on construit, mais la manière dont on décide de le construire.

Avant de construire

L'erreur la plus chère n'est pas un bug. C'est six mois passés à construire quelque chose dont personne ne veut. Ce chapitre décrit les étapes qui coûtent quelques heures et en économisent des centaines.



ÉTAPES COUVERTES : 01 → 04

16 ÉTAPES

Chercher la douleur avant d'avoir l'idée

Si vous n'avez pas encore d'idée, c'est une bonne position, meilleure que d'en avoir une à laquelle vous tenez déjà.

/pain-point-finder part d'un secteur, d'un pays ou d'un métier, et cherche des problèmes **documentés** : ce que les gens écrivent eux-mêmes sur les forums et dans les avis, la littérature académique, les rapports institutionnels, et ce que d'autres ont déjà tenté, y compris leurs échecs, qui sont la source la plus instructive.

Chaque douleur remontée porte **au moins deux sources indépendantes, citées et datées**. Jamais une référence inventée : quand la preuve est mince, le rapport le dit au lieu de combler le trou. Une opportunité fondée sur une source imaginaire est pire qu'aucune opportunité, parce qu'elle a l'air sérieuse.

Passer l'idée au feu

/ideate est un partenaire critique, pas un supporter. Il conteste : le problème est-il réel ou supposé ? Qui exactement souffre ? Que font ces gens aujourd'hui, et pourquoi votre solution serait meilleure que leur système D actuel ? Quel est le modèle économique défendable ? Qu'est-ce qui vous tuerait ?

Il ne valide jamais une idée par politesse. Un « non » argumenté est un succès de cette étape : il vous rend les mois que l'idée allait coûter. Le verdict est un **go**, un **no-go** ou un **pivot**, et il s'appuie sur ce qui a été trouvé, pas sur votre enthousiasme.

Un « non » argumenté est un succès de cette étape : il vous rend les mois que l'idée allait coûter.

Écrire ce qu'on construit

/spec transforme le go en **cahier des charges** : les fonctionnalités, les écrans, ce qui est dans la première version et surtout **ce qui n'y est pas**.

La liste des exclusions est la plus utile du document. Une première version qui essaie de tout faire ne sort jamais, et le harnais vous aidera à défendre ce périmètre quand la tentation d'ajouter reviendra, et elle reviendra.

Ce document n'est pas de la paperasse : **/bootstrap** le lit et construit à partir de lui. Ce que vous écrivez ici devient le projet.

Le modèle économique, si l'argent est en jeu

/bmc déroule les neuf blocs du Business Model Canvas (un outil visuel qui permet de présenter une idée de business sur une seule page) dans leur ordre de dépendance, en partant de ce que l'idéation a validé. Il conteste les blocs faibles (une source de revenu sans segment de clientèle en face, une activité clé que personne ne porte) et esquisse un compte de résultat mensuel dans votre monnaie.

LA RÈGLE DE CE CHAPITRE

Ces étapes produisent des documents, pas du code. C'est exactement pour ça qu'elles sont rentables : **un document se jette sans regret, du code qu'on a payé, non.**

À RETENIR

Quatre étapes, zéro ligne de code. Elles ne servent pas à préparer le travail : elles servent à savoir s'il faut le faire.

Ce qui tourne sous le capot

Vous n'avez pas besoin de savoir programmer. Mais savoir quelles pièces composent votre produit vous sert trois fois : pour comprendre ce que vous payez, pour parler à un prestataire si un jour vous en prenez un, et pour ne pas être prisonnier d'un outil.

Ce chapitre est le seul du livre à nommer des technologies. Vous pouvez le sauter en première lecture.

Les pièces, et à quoi chacune sert

Le site lui-même	Next.js
Il produit des pages rapides et bien référencées, et fonctionne aussi bien sur téléphone que sur ordinateur, et le harnais l'exige à la même sévérité sur les trois tailles d'écran, ce qui n'est pas un détail dans un marché où la majorité des visiteurs arrivent par mobile.	
La base de données	PostgreSQL, chez Neon
C'est là que vivent vos comptes utilisateurs, vos commandes, votre contenu. Facturée à l'usage, avec un palier gratuit qui suffit largement à démarrer.	
L'hébergement	Vercel
Votre site y est publié, mis à jour à chaque changement, et servi depuis une région choisie selon votre marché. Cette dernière précision vaut de l'argent : une base à Francfort et un serveur à Paris se parlent plusieurs fois par page affichée, et le harnais les apparie pour éviter cette latence payée en silence.	
Les paiements	Stripe, ou Fedapay en zone UEMOA
Un fournisseur adapté à votre marché : Fedapay accepte le mobile money que vos clients utilisent réellement.	
Les emails	Resend ou Brevo
Confirmations de compte, réinitialisations de mot de passe, notifications.	

FIGURE 4 : LES PIÈCES DE VOTRE PRODUIT, ET À QUOI CHACUNE SERT

Ce que ça coûte pour démarrer

Presque tout ce qui précède a un palier gratuit, et le harnais est conçu pour y rester le plus longtemps possible. Les commandes `/quotas` et `/billing` vous montrent où vous en êtes sur chaque

service, et vous alertent **avant** la falaise plutôt qu'après.

La première dépense n'est pas dans cette liste. Pour exécuter ces commandes, il faut un accès à l'assistant qui les exécute, et deux portes existent. **Antigravity**, l'hôte de Google : son palier gratuit s'utilise tel quel, sans abonnement, et il comporte une limite hebdomadaire que Google ne publie pas. **Claude Code**, l'hôte d'Anthropic : par abonnement, par compte facturé à l'usage (souvent la voie la moins chère pour commencer), ou via un fournisseur tiers. Le harnais tourne sur les deux, mêmes commandes, mêmes projets. Si la dépense compte, commencez par la porte gratuite.

CE QUE ÇA COÛTE : L'AVERTISSEMENT JURIDIQUE

Un plan gratuit n'autorise pas forcément un usage commercial.

[/commercial-check](#) lit les conditions de chaque service que votre projet utilise et vous dit ce qu'il faut régler avant de vendre. Il échoue fermé : une condition non vérifiée n'est *jamais* présentée comme un feu vert.

Le mot qui compte : vous n'êtes pas enfermé

Tout ce que le harnais construit vous appartient. Le code est dans votre dépôt, la base est sur votre compte, le nom de domaine est à votre nom. Si vous décidez un jour de continuer sans le harnais, ou avec quelqu'un d'autre, **rien ne vous retient** : c'est du code standard que n'importe quel développeur du métier sait reprendre.

À RETENIR

Cinq pièces, toutes standard, toutes à votre nom. Le palier gratuit suffit pour démarrer, mais vérifiez qu'il autorise de vendre avant de vendre.

Le démarrage du projet

/bootstrap est la commande qui transforme un dossier vide en projet en ligne. En une séance, vous passez d'une idée écrite à une adresse que vous pouvez ouvrir dans un navigateur et montrer à quelqu'un.



ÉTAPE COUVERTE : 05, ÉCHAFAUDAGE ET DÉPLOIEMENT

16 ÉTAPES

Ce qu'il vous demande, et pourquoi

Il pose peu de questions, et chacune décide de quelque chose qu'on ne peut pas deviner.

Q1 Ce que vous construisez

Si vous avez un cahier des charges, il le cherche et le lit au lieu de vous refaire raconter le projet.

Q2 Pour quel marché

Le pays et la monnaie décident du fournisseur de paiement proposé, de la région d'hébergement, des canaux marketing réalistes plus tard. Un projet togolais et un projet français ne se construisent pas pareil, et le deviner mal est pire que demander.

Q3 Site ou application web, ou application mobile

Posée en premier, parce que toute la suite en dépend. Deviner mal produirait un site ou une application web pour quelqu'un qui voulait une application mobile, chaque étape annonçant un succès.

Q4 Qui publie le projet

La raison sociale, l'adresse du siège, le responsable de publication. Ces informations partent dans vos mentions légales, un document opposable.

FIGURE 5 : LES QUATRE QUESTIONS DU DÉMARRAGE, ET CE QU'ELLES DÉCIDENT

Si la machine a déjà servi à publier pour une entreprise, elle vous la **propose**, et vous la confirmez à voix haute, même s'il n'y en a qu'une. Cette insistance a une raison : redemander huit champs coûte deux minutes, tandis que publier le numéro d'immatriculation d'un client sur le site d'un autre est une faute dans un document légal, et une faute **invisible**, car personne ne relit ses mentions légales après la mise en ligne.

Ce qu'il fait sans le demander

- Le dépôt de code créé et privé par défaut
- Le socle du site posé
- La base de données provisionnée dans la bonne région
- Le site déployé et vérifié, non pas « la commande a répondu » mais le déploiement est allé au bout et la page répond
- Les pages légales écrites, avec la loi de votre pays et non celle d'un autre continent
- Le plan du site, le référencement de base, le fichier qui dit à Google quoi indexer
- La licence et l'attribution du projet, dès le premier commit

SEPT CHOSES FAITES SANS QU'ON LES DEMANDE

Ce dernier point est celui qu'on remarque seulement quand il manque : si vous vendez ce projet à un client, il est en règle le jour de la livraison.

À la fin

Vous avez une adresse. Elle affiche une page d'accueil sobre, elle est en ligne, elle est à vous. Ce n'est pas encore votre produit : c'est le socle sur lequel les chapitres suivants ajoutent ce qui le rend utile.

Le harnais vous annonce alors l'étape suivante et sa raison, et vous n'avez qu'à la suivre.

Ce que la question mobile décide

Une précision, parce qu'elle décide de votre architecture. Le harnais construit des sites et des applications web complets : **104 de ses 105 commandes y tournent**. Du côté mobile, la vraie question n'est pas « mobile ou web » : c'est **ce que vous vendez**.

Les boutiques d'applications distinguent deux choses. Ce qui est livré dans le monde réel, une course, une livraison, un repas, une marchandise, une prestation, ne relève pas de leur système d'achat intégré : votre application encaisse librement, par T-Money, Flooz ou autre. C'est ainsi que Gozem vous fait payer une course. Ce qui est consommé *dans* l'application, un abonnement, des crédits, du contenu premium, doit au contraire passer par leur système d'achat intégré, lequel n'accepte pas le mobile money. Là, et là seulement, la porte est fermée : pour notre marché ce n'est pas une commission de 30 %, c'est un moyen de paiement qui n'existe pas chez eux.

LIVRÉ DANS LE MONDE RÉEL

Une course, une livraison, un repas, une marchandise, une prestation.
Votre application encaisse librement, par mobile money.

CONSOMMÉ DANS L'APPLICATION

Un abonnement, des crédits, du contenu premium. Passage obligé par leur système d'achat intégré, **qui n'accepte pas le mobile money.**

FIGURE 6 : LA DISTINCTION QUI DÉCIDE DE VOTRE ARCHITECTURE

D'où la solution que le harnais applique par défaut quand vous vendez du numérique : on encaisse sur le web, là où le mobile money passe sans obstacle, et l'application se contente de

consommer un droit déjà payé. Vous n'avez rien à arbitrer, c'est le mode proposé d'office.

Et une limite qui est la nôtre, celle-là

Aujourd'hui, sur un projet mobile, le harnais crée l'application, son serveur en ligne et les comptes de vos utilisateurs : cette partie existe et a été éprouvée. Ce qu'il ne sait pas encore faire à votre place, c'est brancher la caisse. L'encaissement est écrit pour un site ou une application web, et personne ne l'a encore relié à un écran d'application mobile. Vous pouvez donc démarrer un projet mobile avec le harnais ; le paiement, lui, se câble encore à la main.

CE CHANTIER EST ENGAGÉ

Rendre le harnais aussi complet sur mobile que sur le web est au programme, et l'encaissement en est la première tranche. Nous ne donnons pas de date : ce harnais a pour règle de n'annoncer une capacité qu'après l'avoir vue fonctionner pour de vrai. C'est cette règle qui vous protège, et elle vaut aussi quand elle nous ralentit. Vous n'aurez pas à guetter l'annonce : les mises à jour vous parviennent avec votre accès, et la commande `/about` vous dit à tout moment ce que VOTRE version sait faire sur un site ou une application web et sur une application mobile. Le jour où cette page sera périmée, c'est `/about` qui vous le dira.

À RETENIR

Une séance, quatre questions, une adresse en ligne. Ce qui vous est demandé est exactement ce qui ne peut pas être deviné ; tout le reste est fait sans vous le demander. Et sur un projet mobile, ce que vous vendez décide de tout : le réel s'encaisse librement, le numérique s'encaisse sur le web.

Les outils du quotidien

Une fois le projet né, quelques commandes reviennent presque chaque séance. Les connaître change le rythme de travail.



ÉTAPE COUVERTE : 06, IDENTITÉ ET LANDING

16 ÉTAPES

Savoir où vous en êtes

/next

Répond à « et maintenant ? ». Il ne se fie pas à ce que le projet déclare avoir fait : il regarde ce qui existe réellement (les documents sur le disque, la base, la branche de test, le déploiement) puis recommande une étape avec sa raison.

Donner un visage au produit

/design

Fixe l'identité visuelle : couleurs, typographie, ton. Il écrit ces choix dans le projet, ce qui évite que chaque nouvelle page réinvente une apparence.

/brand-kit

Va plus loin : logo, couvertures pour les réseaux, gabarits de publication, et surtout les consignes de production de chaque visuel, prêtes à être exécutées.

/generate

Produit les fichiers eux-mêmes. La seule famille qui dépense de l'argent réel : elle annonce le prix avant chaque production, ne dépense rien sans confirmation explicite, télécharge immédiatement, et écrit à côté de chaque fichier un journal disant quel modèle, quel prix, quelle consigne.

/montage

Assemble et finit : recadrage, texte incrusté, assemblage de plans vidéo. Cette moitié-là est gratuite : c'est du traitement local.

Vérifier avant de montrer

/security

Audite : secrets exposés, routes non protégées, validation des entrées, dépendances vulnérables, en-têtes. Son verdict est enregistré, et la mise en ligne le consulte.

/seo · /seo-perf · /eco-audit

Le référencement, la vitesse réelle mesurée chez Google plutôt que supposée, l'empreinte de vos pages.

/rgpd-audit

Tient à jour le registre de vos sous-traitants et la page de confidentialité, un document opposable rédigé avec la loi de votre pays.

Surveiller ce que ça coûte

/quotas

Fait le tour de vos services et dit où vous en êtes sur chaque palier gratuit, avec les falaises nommées, celles dont le franchissement suspend le service plutôt que de le facturer.

/save-project · /clean

Un instantané : base, fichiers, configuration. Et le rangement.

Quand vous ne savez plus

/about

Dit ce qu'est le harnais, quelle version tourne, combien de commandes existent.

/prof

Enseigne : c'est le cours dont ce livre est tiré.

Et **/next**, toujours. C'est la commande à retenir si vous n'en retenez qu'une.

Une dépense sans trace est une dépense qu'on refait par accident.

À RETENIR

Cinq réflexes : demander où vous en êtes, habiller le produit, vérifier avant de montrer, surveiller la facture, et demander de l'aide au harnais lui-même.

Ajouter une capacité

Votre site existe. Il lui manque des comptes utilisateurs, des paiements, un formulaire de contact. Chaque capacité s'ajoute par une commande qui écrit le code, pose la configuration, et met à jour la documentation du projet.

Les grandes familles

Les comptes /add-auth /add-google-auth · /add-github-auth · /add-2fa · /add-role	L'argent /add-fedapay /add-stripe · /billing
Les données /add-db /db-isolate · /add-backup-db	Le contact et la présence /add-email /add-domain · /new-email-address
Le confort /add-i18n /add-dark-mode · /add-pwa · /add-map · /add-storage · /add-analytics	L'automatique /add-cron /add-workflow · /add-agent

FIGURE 7 : LES SIX FAMILLES DE COMMANDES D'AJOUT

Pour les comptes, **/add-auth** installe l'authentification. Par défaut il crée des comptes utilisateurs ET un administrateur, parce qu'un produit avec des utilisateurs et sans personne pour les administrer

est un produit qu'on découvre incomplet des semaines plus tard, une fois l'application bien avancée.

Pour l'argent, `/add-stripe` couvre les cartes internationales et `/add-fedapay` le mobile money d'Afrique de l'Ouest. La bascule en mode réel, celle où de vrais euros ou francs circulent, exige toujours votre accord explicite.

Pour les données, `/db-isolate` sépare votre base de travail de la base réelle : c'est l'étape du chapitre suivant, et la sauter est la faute la plus coûteuse du parcours.

Pour l'automatique, la différence entre un enchaînement (workflow) et un agent n'est pas l'intelligence, c'est la **boucle** : un enchaînement fini se termine, un agent tourne.

Ce qu'elles font toutes, sans le dire

Chacune vérifie que ce dont elle dépend est déjà là, refuse proprement en le nommant sinon, écrit sa configuration aux bons endroits, met à jour la documentation du projet, et déclare ses nouveaux sous-traitants au registre de confidentialité quand elle en introduit.

Ce dernier point est celui qu'on oublierait toujours à la main : ajouter une carte interactive envoie l'adresse IP de chaque visiteur chez un tiers, et votre page de confidentialité doit le dire.

LA RÈGLE DE CE CHAPITRE

Une capacité à la fois. Ajoutez, regardez le résultat, validez. Trois capacités ajoutées d'un coup produisent un projet dont vous ne savez plus laquelle a cassé quoi.

À RETENIR

Six familles, une grammaire commune : chaque commande vérifie ses dépendances, écrit sa configuration, et déclare ce qu'elle vient d'exposer à un tiers.

Mettre en ligne, en trois temps

Voici le chapitre où l'on perd de l'argent quand on va trop vite.



Trois environnements, toujours



FIGURE 8 : LES TROIS ENVIRONNEMENTS, ET LES DEUX PORTES ENTRE EUX

Pourquoi trois et pas deux

Parce que le premier test de paiement doit être fait par vous avec de l'argent factice, et non par votre premier client avec le sien.

Et parce qu'une base de test vous laisse essayer les cas laids (créer, supprimer, refaire) sans risquer les données de gens réels.

Un test sur la base de production n'est pas un test, c'est une manipulation en direct.

Ce que le harnais vérifie à votre place

Sur le palier de test comme en production, il ne se contente pas de lancer le déploiement : il **vérifie que la construction a réussi**, sur la bonne branche. Cette précision a une histoire : demander « le dernier déploiement de test » sur un projet à plusieurs branches vivantes répond sur le déploiement de quelqu'un d'autre, et ce verdict ressemblerait à un succès tout en ne prouvant rien.

Il vérifie aussi qu'une adresse de test n'est pas derrière un mur de connexion : une copie privée que vous seul pouvez ouvrir ne sert pas à faire tester quelqu'un.

La séquence, en clair

- 01 Travailler et tester en **local**.
- 02 Enregistrer le travail et vous le montrer, pour validation.
- 03 Pousser sur la branche de **test**, et vous inviter à l'essayer en conditions réelles.
- 04 **Seulement après votre accord explicite**, publier en production.

Jamais de raccourci vers l'étape 4. Vous pouvez configurer le harnais pour qu'il soit plus autonome sur les étapes 2 et 3 ; l'étape 4 reste la vôtre, sauf à l'avoir explicitement demandé autrement.

Et l'autonomie ne s'achète pas d'un simple réglage. Même au niveau le plus permissif, le harnais exige en plus une **preuve de sécurité fraîche** et une vérification de code sans avertissement. Déléguer davantage ne veut pas dire vérifier moins.

La dernière porte

/launch est la vérification d'avant-ouverture : l'audit de sécurité est-il passé, les valeurs d'exemple ont-elles été remplacées, les conditions commerciales des services que vous utilisez sont-elles réglées. Il bloque sur ce qui doit bloquer.

Et il ne se contente pas de trouver un audit réussi quelque part dans l'historique. Un audit passé sur du code que quarante enregistrements ont depuis remplacé ne prouve plus rien : la fraîcheur se juge sur l'**empreinte du contenu vérifié**, et non sur une date. Si ce n'est plus le même code, l'audit est refait.

Le reste, le nom de domaine et les emails à ce nom, se pose avant ou après, mais la vérification, elle, passe avant.

À RETENIR

Local pour casser, test pour essayer comme un client, production pour vendre. Deux portes entre les trois, et la seconde ne s'ouvre que de votre main.

Trouver des clients

Un produit en ligne que personne ne connaît ne rapporte rien. Cette moitié du parcours est celle qu'on saute le plus souvent, et c'est celle qui décide si le reste aura servi.



ÉTAPES COUVERTES : 11 → 15, LA PISTE MARCHÉ

16 ÉTAPES

Le plan, avant les publications

/gtm construit le plan d'acquisition de **votre** produit, pas du marketing générique. Il lit ce que l'idéation a validé et le marché déclaré du projet, puis travaille quatre choses :

01 Qui signe en premier Pas « les PME » : la personne précise qui a le problème, celle qui paie, celle qui peut bloquer. Ces trois-là sont parfois trois personnes différentes, et le message n'est pas le même.	02 La proposition de valeur en une phrase Dans les mots du client, pas dans les vôtres.
03 Les canaux réalistes de votre marché Avec un coût et un effort par canal. Ce qui marche à Lomé n'est pas ce qui marche à Paris, et le harnais ne recommande pas un canal que vos clients n'utilisent pas.	04 Un plan daté pour les cent premiers utilisateurs Avec des objectifs hebdomadaires et un budget dans votre monnaie.

FIGURE 9 : LES QUATRE LIVRABLES DU PLAN D'ACQUISITION

Puis le visage, puis les mots

L'ordre compte, et il a été appris à ses dépens : **/gtm** d'abord, **/pitch** ensuite, **/brand-kit** en dernier. Un kit visuel construit avant le plan produit des visuels pour des canaux que le plan écartera. C'est arrivé, et c'était payé à l'image.



FIGURE 10 : L'ORDRE, ET IL N'EST PAS INTERCHANGEABLE

/pitch écrit l'argumentaire : la page de vente, le message de prospection, la présentation courte. */brand-kit* habille ensuite ce qui a été décidé.

Exécuter

/promote transforme le plan en matériel prêt à lancer : séquences de messages avec la discipline anti-spam intégrée, calendrier de contenu, briefs de campagne publicitaire avec ciblage et budget, plan de mesure.

CE QUE ÇA COÛTE : LA DÉPENSE PUBLICITAIRE

/promote propose les campagnes ; c'est vous qui les lancez. Toute dépense publicitaire passe par votre accord.

Mesurer, et corriger

/metrics installe le suivi : combien de visiteurs, combien s'inscrivent, combien paient, et où ils abandonnent. Sans ces chiffres, améliorer un produit revient à deviner.

L'endroit où les gens abandonnent est presque toujours une surprise. C'est pour ça qu'on mesure au lieu de supposer.

Le meilleur moment pour savoir qui achètera est avant d'avoir fini de construire, car il est encore temps de changer ce qu'on construit.

À RETENIR

La piste marché avance en parallèle de la construction, pas après. Le plan d'abord, les mots ensuite, le visuel en dernier, et la mesure pour ne plus supposer.

Bien décrire ce qu'on veut

C'est la compétence centrale de cette manière de travailler, et la seule que le harnais ne peut pas exercer à votre place.

ÉTAPE COUVERTE : 04, LE CAHIER DES CHARGES

16 ÉTAPES

Le principe : décrire le résultat, pas la méthode

UNE BONNE DEMANDE

« Fais une page où les clients voient leurs commandes passées, de la plus récente à la plus ancienne, avec le statut de chacune. »

UNE MAUVAISE DEMANDE

« Crée un composant React avec un tableau trié. »

FIGURE 11 : LA MÊME PAGE, DEUX DEMANDES

La seconde n'est pas mauvaise parce qu'elle est technique, mais parce qu'elle **impose une solution** avant d'avoir décrit le besoin. Si le tableau n'était pas la bonne forme sur téléphone, vous venez d'interdire la bonne réponse.

Dites **quoi** et **pour qui**. Laissez le comment.

Les quatre questions qui rendent une demande exécutable

Avant de demander quelque chose, vérifiez que votre phrase répond à ceci :

QUI Qui utilise cette fonctionnalité ? Un visiteur, un client connecté, un administrateur ? La réponse décide des protections.	QUOI Que doit-il pouvoir faire ? Concrètement, en une action nommée.
QUAND Quand cela peut-il mal se passer ? Une commande annulée, un paiement refusé, un champ vide. Les cas laids sont ceux qu'on oublie de décrire, donc ceux qui manquent au résultat.	COMMENT Comment saurez-vous que c'est réussi ? Si vous ne pouvez pas décrire la vérification, la demande n'est pas encore assez précise.

FIGURE 12 : LES QUATRE QUESTIONS, À PASSER AVANT CHAQUE DEMANDE

Reconnaître un résultat, sans lire le code

Vous ne relisez pas le code. Vous vérifiez le **comportement**, et cela s'apprend vite : essayez le cas normal, puis **essayez de casser**. Champ vide, texte à la place d'un nombre, double clic sur le bouton d'achat. Regardez sur téléphone, pas seulement sur ordinateur, car la majorité de vos visiteurs y sera.

Et si quelque chose vous paraît étrange, dites-le, même sans savoir l'expliquer. « Cette page met du temps à s'afficher » est un rapport parfaitement utile.

Ce qu'il ne faut pas faire

Ne validez pas ce que vous n'avez pas regardé. Un « ça a l'air bon » sur un écran jamais ouvert est la manière la plus sûre de découvrir un défaut chez un client.

Ne demandez pas trois choses dans une phrase. Vous ne saurez pas laquelle a mal tourné.

Ne dites pas « comme d'habitude ». Chaque conversation repart d'un contexte propre ; ce qui a été décidé se réécrit ou se range dans le projet, et le harnais a justement un manifeste pour ça, qui retient une décision une fois pour toutes.

LE RACCOURCI

Si tout cela vous paraît beaucoup : `/spec` pose ces questions à votre place, dans l'ordre, et écrit le document. Vous n'avez qu'à répondre.

À RETENIR

Décrivez le résultat, pas la méthode. Quatre questions (qui, quoi, quand ça casse, comment on vérifie) suffisent à rendre presque n'importe quelle demande exécutable.

Travailler avec la machine

Quelques habitudes qui séparent une séance productive d'une séance frustrante.

Une chose à la fois

Demandez une modification, regardez-la, validez-la. Puis la suivante.

Trois demandes empilées produisent un résultat où vous ne savez plus quoi a cassé quoi, et revenir en arrière devient une négociation. Le harnais enregistre le travail par petits paliers pour cette raison exacte : un palier validé est un point de retour.

Dire ce qui ne va pas, sans savoir pourquoi

Vous n'avez pas à diagnostiquer. « Le bouton ne fait rien quand je clique deux fois vite » est un excellent rapport, meilleur qu'une hypothèse sur la cause, qui enverrait chercher au mauvais endroit.

01 Ce que vous avez fait	02 Ce que vous attendiez	03 Ce qui est arrivé
-----------------------------	-----------------------------	-------------------------

FIGURE 13 : LES TROIS PHRASES QUI SUFFISENT PRESQUE TOUJOURS

Les moments où on vous demande de choisir

Le harnais s'arrête à certains points et attend. Ce n'est pas de l'hésitation : ce sont les décisions qui vous appartiennent.

Et ces arrêts ne sont pas une intention écrite dans une consigne, que la machine pourrait interpréter largement un jour de zèle. Ils sont tenus par un **registre de vingt-quatre portes** que le harnais doit consulter avant d'agir : l'argent réel, les suppressions, les écritures en production, le passage d'un paiement en mode réel. Une règle qu'on peut contourner par une formulation habile n'est pas une règle.

L ' ARGENT	Toute dépense réelle, toute bascule d'un paiement en mode réel.
LA DESTRUCTION	Supprimer une base, un projet, un dépôt.
LA PRODUCTION	Publier sur le site public.
LE MODÈLE	Certaines étapes recommandent un modèle plus capable, donc plus cher, parce que l'enjeu le justifie. Le harnais vous le signale, vous basculez si vous le voulez, et il continue.

FIGURE 14 : LES QUATRE ARRÊTS OÙ LA DÉCISION EST LA VÔTRE

Répondez à ces questions plutôt que de les contourner. Elles existent parce que quelqu'un a payé leur absence.

Quand la conversation devient longue

Une longue séance perd en précision : les décisions du début s'estompent. Le harnais le signale et propose une **passation**, un document qui résume l'état, les décisions prises et leurs raisons, ce qui reste à faire, plus un texte prêt à coller pour reprendre ailleurs. La commande **/handover** sert à cela.

Acceptez quand c'est proposé. Une passation s'écrit pendant que le contexte est encore net, pas quand il ne l'est plus.

Ce que la machine ne fera pas à votre place

Elle ne sait pas si votre prix est le bon. Elle ne sait pas si vos clients préfèrent le mobile money à la carte. Elle ne sait pas si le nom que vous avez choisi sonne mal dans la langue de votre marché.

Elle construit vite et bien ce que vous décidez. **La décision reste la partie humaine**, et c'est celle qui fait la différence entre un produit qui marche et un produit qui existe.

À RETENIR

Une chose à la fois, trois phrases pour signaler un problème, quatre arrêts qui vous appartiennent. Le reste, la machine le fait.

POUR FINIR

Vous avez la carte

Le reste est du travail, et le travail commence par une commande.

/pain-point-finder

Si vous n'avez pas encore d'idée.

/ideate

Si vous en avez une, et acceptez qu'elle
soit contestée.

/next

Si vous êtes déjà en route et perdu, à n'importe
quelle étape.

La différence entre les gens qui livrent et les autres n'est pas le talent technique. C'est d'avoir commencé, puis de ne pas s'être arrêté à la première étape ennuyeuse, qui est souvent l'audit de sécurité, ou le plan d'acquisition, c'est-à-dire précisément celles qui décident si le produit vivra.

POUR ALLER PLUS LOIN

Par où commencer

Ce livre décrit le chemin. Le parcourir seul est possible : le harnais est fait pour ça, et quand vous ne saurez plus quoi faire, /next vous dira l'étape suivante et pourquoi c'est celle-là.

Quatre portes, si vous préférez ne pas commencer seul.

Une séance de prise en main

À Lomé ou dans une autre ville. Une journée en petit groupe : l'outil est présenté, expliqué, puis **installé sur votre propre poste** avec l'aide de l'auteur de ce livre. Vous repartez avec le harnais qui tourne chez vous. Les prochaines dates et les places disponibles sur dickoindustries.com/soka

Le bootcamp Dicko Soka

Un ou deux jours pendant lesquels vous travaillez sur **votre** projet jusqu'à sa mise en ligne, avec le coaching de l'auteur. À l'issue, votre application est en ligne : la vôtre, pas un exercice. Les prochaines dates et les places disponibles sur dickoindustries.com/soka

La formation en ligne

Pour qui ne peut se déplacer. Même méthode, même parcours, même exigence. Écrivez-nous : nous prévenons en priorité ceux qui se sont manifestés. contact@dickoindustries.com

Une question, simplement

WhatsApp **+228 96 64 92 92**. C'est l'auteur de ces pages qui répond.

Vous n'avez besoin d'aucune des quatre pour commencer. Vous avez la carte.



À propos de cette édition

Ce livre décrit Dicko Soka 3.86.0. Compilé le 18 août 2026.

Tous les faits chiffrés de cet ouvrage (le nombre de commandes, la liste et l'ordre des 16 étapes du parcours, les noms des commandes citées) ont été **lus dans le harnais au moment de la compilation**, et non recopiés à la main. Un contrôle automatique refuse de produire ce livre si une commande qu'il nomme n'existe pas, si une étape du parcours n'est pas traitée, ou si un chiffre y est écrit en dur.

Cela ne rend pas ce livre éternel : cela garantit qu'il était **exact pour la version nommée ci-dessus**. Le harnais a connu trois versions en une seule journée. Si la vôtre est plus récente, la commande **/about** vous dira laquelle vous avez, et une édition à jour de ce livre peut être refabriquée à partir des mêmes sources.

64

COMMANDES PUBLIQUES

41

ROUAGES INTERNES

105

AU TOTAL

16

ÉTAPES AU PARCOURS

Mohamed Nassirou Édité par Dicko Industries

Dicko Soka · Dicko Tools · TransiFa Docs

Fondateur de Dicko Industries

ÉDITÉ PAR DICKO INDUSTRIES

DICKO SOKA · DICKO TOOLS · TRANSIFA DOCS