



Think Tank – Do Tank

sur l'IA Générative appliquée au domaine de l'Ingénierie Logicielle

3ème édition – Mercredi 26 janvier

Table Ronde : " Une remise en cause du modèle SDLC ? "

Réunion Générale - 3ème édition

Table Ronde : " Une remise en cause du modèle SDLC ? "



René JULLIEN : Head of Engineering Enablement
Strategy, Portfolio and Incubation

AMADEUS



Philippe MARTIN : Group CTO

bpifrance



Pascal FAUCILLON : Directeur des Opérations

 **EDF**



Julien LEPINE : Group CTO



35 minutes

L'ÈRE DE LA DSI AUGMENTÉE REPENSER LE MODÈLE OPÉRATIONNEL À L'HEURE DE L'IA AGENTIQUE



PHILIPPE MARTIN
GROUP CTO

La vague IA agentique

Promesses, vitesse et incertitude -> anticipation des changements



La réduction radicale de la taille des équipes

La réduction de la taille des équipes ne signifie pas faire moins mais faire autrement.



On peut passer d'une équipe de 8 personnes à 4 équipes de 2 personnes car **le paradigme du travail change : de l'exécution manuelle à l'orchestration à haute vitesse**

Le nouveau paradigme : de l'exécution à l'orchestration

Hier : L'équipe de maçons



Un architecte et 7 ouvriers spécialisés (maçons, électriciens, plombiers). La valeur réside dans l'effort 'musculaire' : poser chaque brique, tirer chaque câble, vérifier chaque tuyau.

Coordination manuelle

Demain : L'équipe d'architectes augmentée

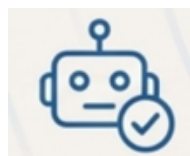
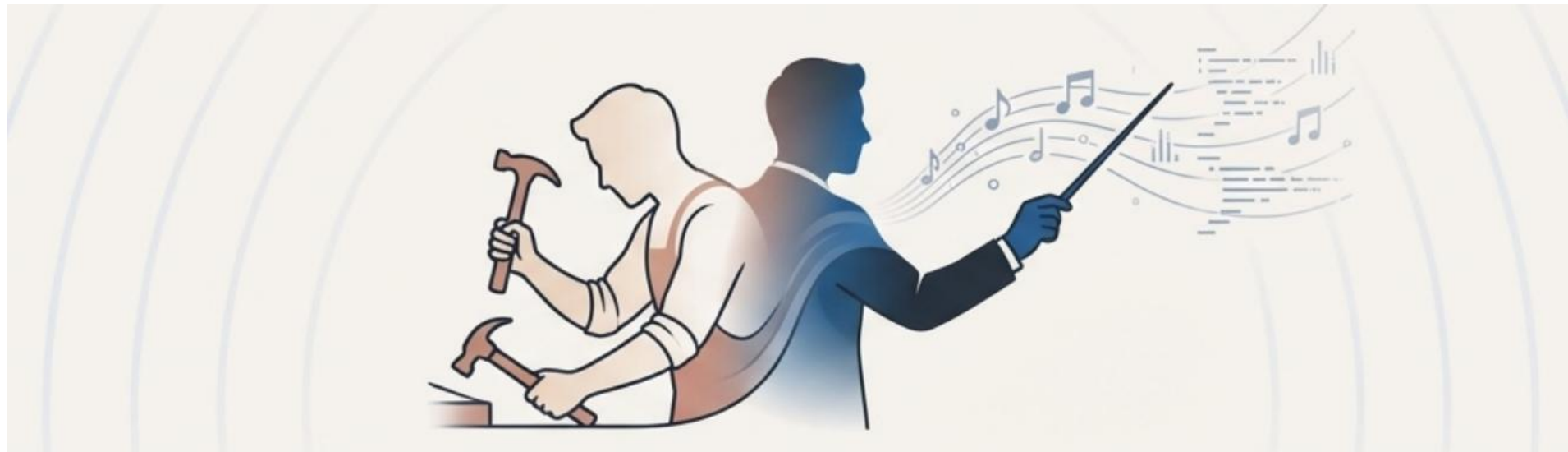


Deux architectes aux commandes d'une flotte de robots autonomes (agents IA). L'un conçoit les plans, l'autre vérifie la conformité du travail fini.

Orchestration automatisée

Le développeur devient chef d'orchestre

Le rôle du développeur évolue radicalement. Il ne produit plus la matière première (le code), il dirige ceux qui la produisent (les agents IA). **Sa valeur se déplace de la production à la conception / supervision.**



Autonomie de l'agent

L'agent ne fait pas qu'aider, il exécute des tâches complètes



Abstraction du travail

L'humain remonte dans la chaîne de valeur, de la production à la conception



Densité de talent

Les 2 personnes restantes doivent être des profils très sénior (type architecte) car elles pilotent une machinerie puissante

Le changement d'activités du développeur : de l'écriture vers l'architecture



Du « comment » au « quoi » (architecture vs syntaxe)

Hier : Maitriser les subtilités d'un framework, optimiser une boucle, ..

Demain : Maitriser le System Design. Savoir décomposer une tâche complexe en sous tâches pour des agents, définition des interfaces.



L'orchestration d'agents (Prompt engineering avancé)

Hier : Ecrire un prompt simple

Demain : Créer des chaines de raisonnement et des boucles de feedback pour corriger les agents. Configurer le contexte, les contraintes et le personae des agents



L'audit et la revue de code

Hier : Relire le code d'un pair

Demain : Scanner un volume de code généré pour y déceler des failles logiques ou de sécurité sans avoir écrit une seule ligne de code. La lecture critique devient plus importante que l'écriture.

Les 3 défis à relever pour le développeur : de l'exécutant au gestionnaire critique



Lâcher prise

Défi : Accepter que le code ne soit pas écrit « comme je l'aurais fait »

Shift : Abandonner le perfectionnisme syntaxique pour se concentrer sur le résultat fonctionnel et la robustesse du système.



Esprit critique exacerbé

Défi : L'IA est une menteuse confiante

Shift : Passer d'une posture de création à une posture de validation. Ne jamais faire confiance aveuglément à l'agent.



Vision produit

Défi : Adresser le besoin métier

Shift : Le temps de « pissage de code » est réduit à zéro. La valeur ajoutée de l'humain se déplace vers la compréhension du besoin métier. Le développeur devient PM technique.

Les risques majeurs de cette transformation



L'atrophie des compétences

A force de ne plus coder, on oublie comment ça marche. Le risque est de ne plus savoir déboguer un problème complexe que l'IA ne peut résoudre.



La dette technique invisible

Les agents peuvent générer du code qui fonctionne mais qui est mal structuré. Si le manager d'IA ne valide que le résultat, la codebase peut devenir un monstre ingérable.



Le piège de la vérification paresseuse

Face à une IA qui a raison à 95% du temps, l'humain baisse sa garde. Les bugs les plus importants se cachent dans les 5% d'erreurs subtiles



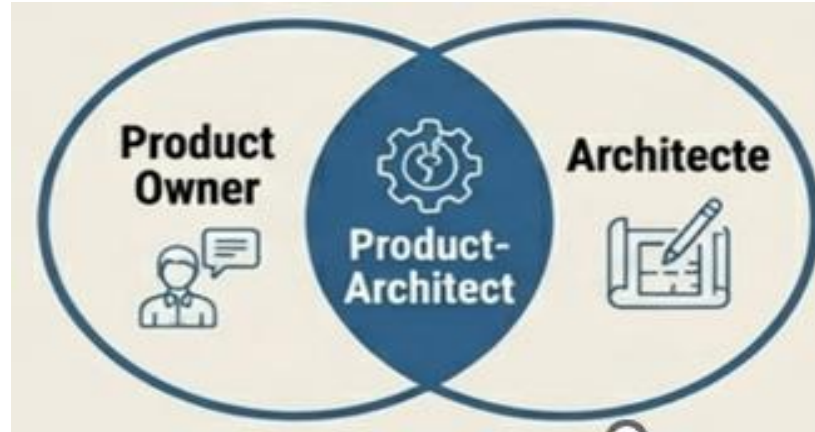
La formation des juniors

On devient sénior car on a beaucoup codé. Si l'IA fait tout le travail, comment les juniors acquerront-ils l'expérience pour superviser l'IA demain ?

Quels impacts au delà du développeur ?

PO

Le problème : Avec les agents produisant le code et l'automatisation des tâches pour le PO, la répartition des activités et rôles de l'Agile Team doivent évoluer.



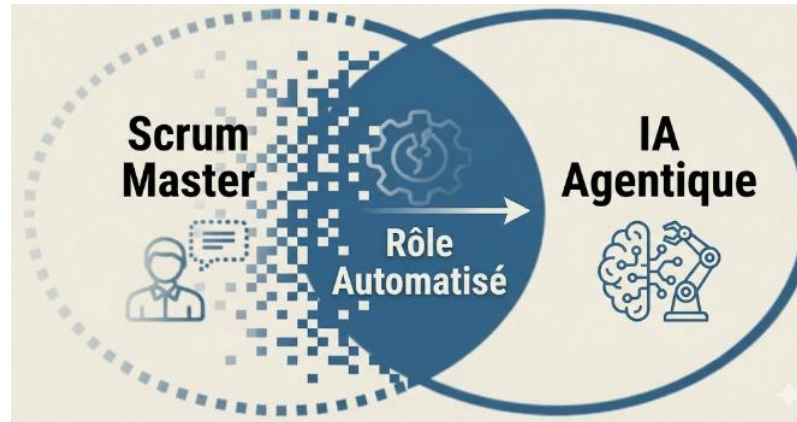
L'évolution :

- Les US sont générées par l'IA sur la base des Intentions produits par le PM.
- L'ex-PO peut produire et challenger un livrable.
- L'ex-développeur a des activités de plus haut niveau : conception & validation.

Ces 2 activités se rejoignent vers un seul et même rôle hybride de Product Architect.

Scrum master

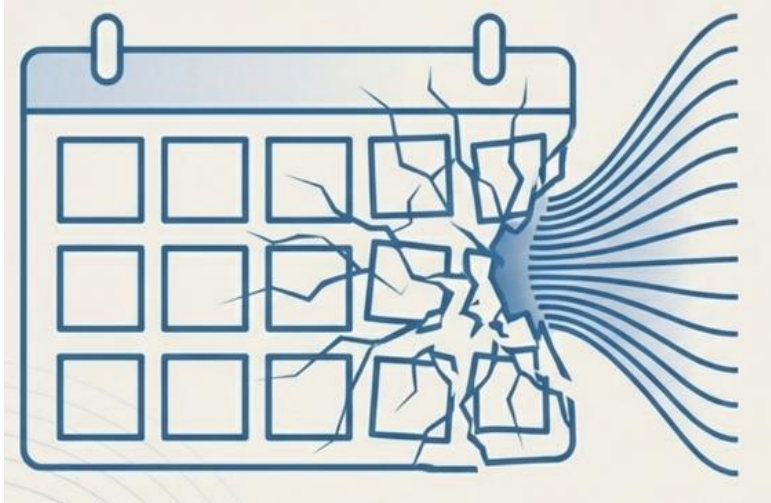
Le problème : Avec une team constituée de 2 personnes il y a beaucoup moins d'interaction à gérer.



L'évolution :

Sans nécessité de synchroniser uniquement 2 personnes, les activités de Scrum Master disparaissent

Impact sur le process agile : La durée des sprints s'effondre



Le sprint de 2 semaines devient techniquement obsolète et inefficace

Si un 'pod' de 2 personnes produit en 3h ce qu'une équipe de 8 produisait en 3 jours, figer le travail dans des blocs de 2 semaines revient à brider la vitesse de production.

Du 'time box' au 'flow' : les nouveaux rythmes de production

Le modèle time boxé : « sprint »

Logique : Rythmer le travail des équipes sur une séquence de développement de durée fixe.

Objectif : Délivrer de la valeur avec un produit fonctionnel



Le bus

Part à heure fixe (toutes les 2 semaines),
qu'il soit plein ou vide



Le taxi

Part dès que le client (la feature) est prêt

Le modèle cible : « Flow » (kanban pur)

Logique : La notion de durée fixe disparaît. Le travail est un flux continu.

Objectif : Dès qu'une tâche est spécifiée, elle est aspirée par les agents. Dès qu'elle est validée, elle part en prod. Les rituels sont déclenchés par l'évènement, non par le calendrier

Conséquences : le « two speed IT »

Process amont : la vision

Temps humain



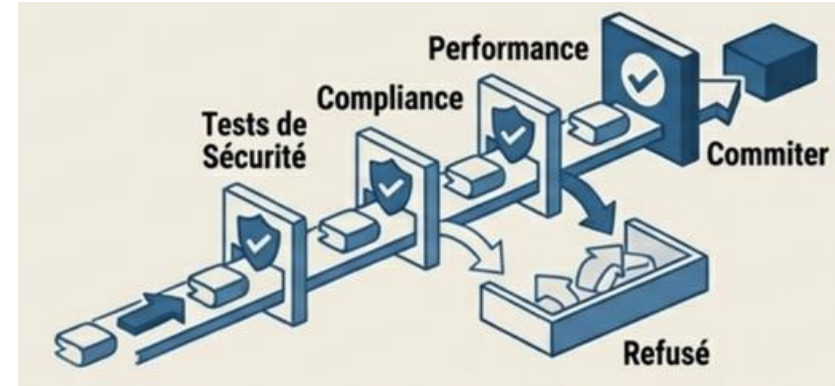
Temps agentique

Temps humain (lent, stratégique) : Cycles de 1 à 2 semaines pour la réflexion produit, l'analyse utilisateur, la **conception** d'architecture.

Temps agentique (instantané) : Cycles d'exécution de quelques heures

Le risque : le goulot d'étranglement n'est plus la capacité à faire mais la capacité à DECIDER quoi faire.

Process aval : qualité



Le problème : L'humain ne peut plus valider tout le volume de code produit. Le goulot d'étranglement se déplace de l'écriture à la validation. Il doit donc s'outiller.

L'évolution : La DOD avec toutes les règles d'architecture, de compliance qui devront être codées. La système team définit et implémente dans les agents ces contrôles. Elle devient la police des agents.

Conséquence : L'estimation change de nature

La mort du poker planning



L'impact : l'estimation en story point devient inutile. Une tâche complexe pour un humain peut prendre 10 min pour un agent.

La nouvelle question : La question n'est plus « combien de temps pour coder ça ? » mais « combien de temps pour vérifier qu' l'IA ne s'est pas trompée ? »

L'effort humain bascule de la production vers la supervision et la validation de la qualité, transformant radicalement le process de développement.

L'onde de choc sur l'agilité à l'échelle (SAFe)



SAFe

Conçu pour synchroniser des masses importantes
(100 à 150 personnes) qui avance LENTEMENT

IA agentique

Conçu pour automatiser des petites unités de 2
personnes (pods) qui avancent à la VITESSE DE
L'ECLAIR

SAFe : structure et rythme évoluent

Impact structurel -> l'effondrement de la taille de l'agile team

Hier : L'atome de base est l'équipe agile (7+/-2).
Un Train 10-12 teams



Train

Agile team

Demain : L'atome de base est le 'pod' (2 ETP)
Un train de 50 pods autonomes
Le rôle de scrum master au niveau de l'équipe disparaît



Train

Flotte de pods

Le train n'est plus un train (qui part à l'heure) mais une flotte de drones (chaque pod est autonome sur sa mission, relié par des API)

Impact sur le rythme -> La fin du PI planning

Le problème : Planifier le travail sur 10 semaines pour des agents capables de coder une feature en 2 heures n'est plus cohérent.
Le backlog est consommé plus rapidement qu'il n'est raffiné.



L'évolution : Le PI planning ne sert plus à planifier les tâches, mais uniquement aligner la vision stratégique et les architectures. On ne planifie plus la production mais la valeur.



Le nouvel operating modèle – synthèse des évolutions

Dimension	Modèle d'hier (exécution manuelle)	Modèle de demain (orchestration agentique)
Equipe	L'agile team (7 +/- 2) - masse critique pour produire	Le 'pod' (2) - densité de talents pour concevoir et superviser
Rôles clé	Le développeur artisan - focus sur le 'comment'	Product-architecte - focus sur le quoi et la validation
Rythme	Le sprint (2 semaines) - time box de production	Le flow (continu) - flux de valeur tiré par le besoin
Echelle	Train SAFe - Synchronisation de masse	Flottes de drones - coordination par API et contrats d'interface

La valeur ne réside plus dans les mains mais dans la vision



On ne paie plus pour l'effort « musculaire » mais pour la vision et la validation

Le développeur demain codera moins, mais challengera plus. Il sera un architecte auditeur.

Réunion Générale - 3ème édition

Table Ronde : " Une remise en cause du modèle SDLC ? "



René JULLIEN : Head of Engineering Enablement
Strategy, Portfolio and Incubation

AMADEUS



Philippe MARTIN : Group CTO

bpifrance



Pascal FAUCILLON : Directeur des Opérations

 **EDF**



Julien LEPINE : Group CTO



L'automatisation du code par l'IA

- Certains l'annoncent pour demain, tandis que d'autres prévoient l'apparition d'un « Super Codeur » en 2031, suivie par le remplacement de toute la chaîne de développement par une « Super IA » d'ici 2034
- Prompting, Agentic et Vibe Coding, une évolution à la vitesse exponentielle qui laisse à penser que n'importe qui peut générer une application en langage naturel ?
- IA, de simple copilote à remplaçant du (junior) développeur ?
Seuls les développeurs expérimentés savent faire un code review du code généré par l'IA ?
Comment un junior peut devenir senior s'il n'a pas pu coder ? « Knowledge Collapse » ?
- Du sujet à la mode à son industrialisation, quelles étapes ...



Réunion Générale - 3ème édition

Table Ronde : " Une remise en cause du modèle SDLC ? "



René JULLIEN : Head of Engineering Enablement
Strategy, Portfolio and Incubation

AMADEUS



Philippe MARTIN : Group CTO

bpifrance



Pascal FAUCILLON : Directeur des Opérations

 **EDF**



Julien LEPINE : Group CTO



11 – 13 février 2001

Manifeste pour le développement Agile de logiciels



With Kiro, I was able to ship more code in the last five months than in the past ten years.

ANTHONY LIGUORI

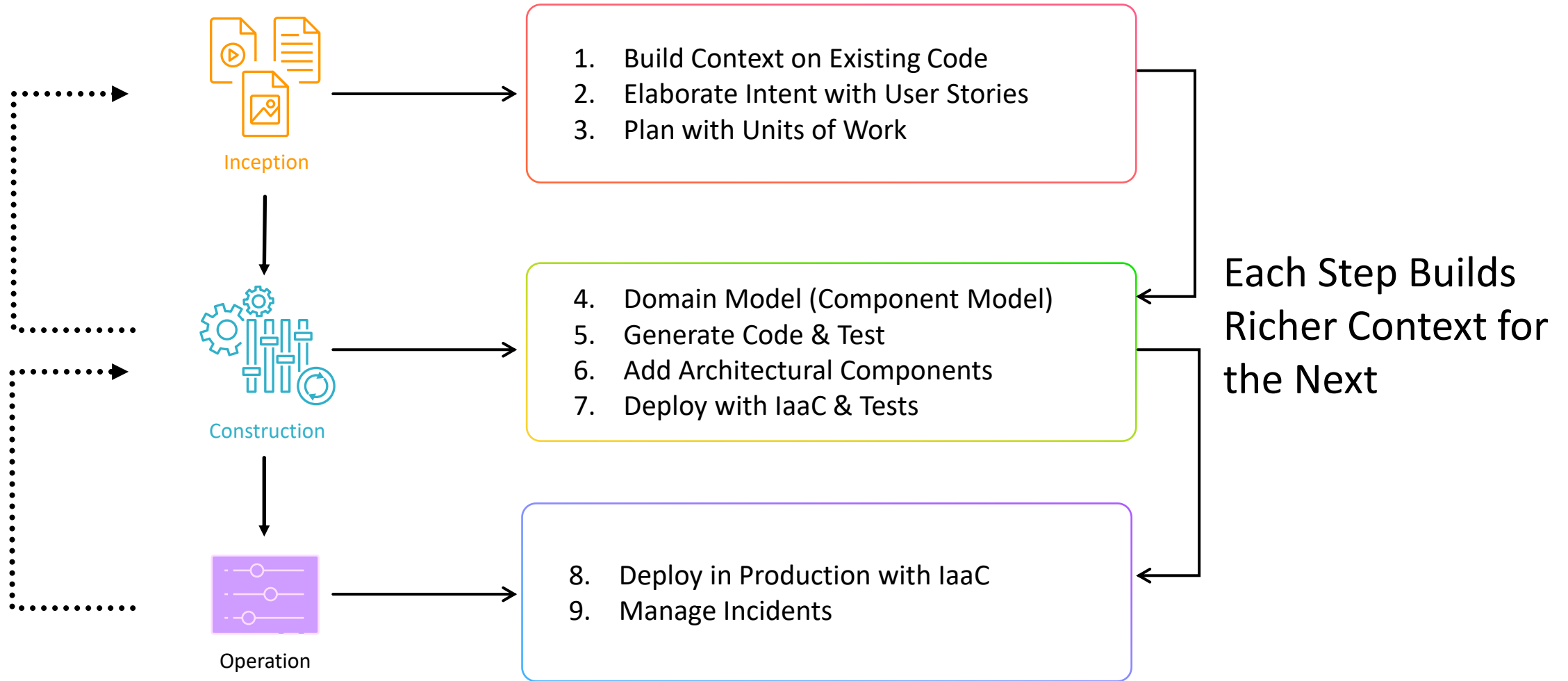
VP, Distinguished Engineer

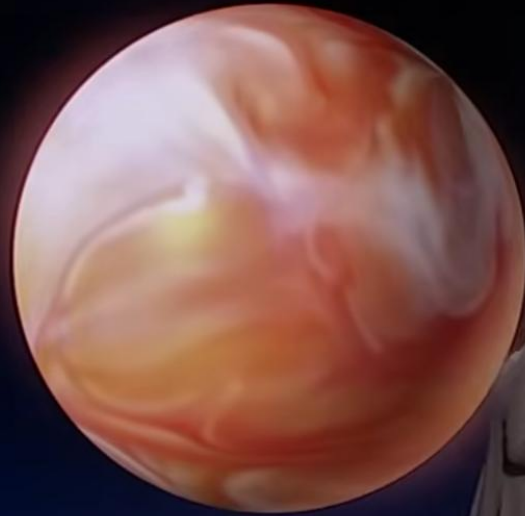


Agents supercharged our developer velocity



AI-DLC, une méthode pour toute l'équipe





Amazon Bedrock Project Mantle

Réunion Générale - 3ème édition

Table Ronde : " Une remise en cause du modèle SDLC ? "



René JULLIEN : Head of Engineering Enablement
Strategy, Portfolio and Incubation

AMADEUS



Philippe MARTIN : Group CTO

bpifrance



Pascal FAUCILLON : Directeur des Opérations

 **EDF**



Julien LEPINE : Group CTO

