Devoxx -Paris 2026

Le palais des congrès a accueilli pendant 3 jours la célèbre conférence autour du développement et de la programmation "DEVOXX PARIS - 2026".

Ayant eu la chance de pouvoir y participer à 5, nous profitons de la sortie des vidéos pour vous proposer, dans cet article, de vous faire un retour sur les différentes nouveautés présentées, mais également sur l'expérience humaine, en espérant vous restituer le mieux possible, la belle aventure que nous y avons vécue.

Afin de vous faire partager au mieux notre expérience, nous allons découper cette présentation en 3 parties :

- L'événement en lui-même
- Les conférences qui nous ont le plus marqués
- Les nouveautés présentes aux stands

Devoxx Kesako?

La DEVOXX est la plus grande conférence sur le développement et la programmation d'Europe. Créée en 2001 en Belgique par Stephan JANSSEN (développeur Java) sous le premier nom de JavaPolis, elle est rapidement devenue la plus grande conférence mondiale indépendante de développeurs Java dans le monde.

Ces conférences sont l'occasion de pouvoir présenter différents sujets liés au développement et à la programmation, comme les bonnes pratiques, ou l'évolution du métier, animés par des spécialistes du sujet.

Aujourd'hui, elle est présente internationalement dans plusieurs pays sous différentes déclinaisons (Belgique, Royaume-Uni, France, Pologne, Maroc, Etats-Unis, Ukraine).

Au niveau de l'organisation, l'événement se fait en général sur 2 à 3 jours, avec des conférences sur différents sujets liés au développement et à la programmation, des séances de pratiques (Hands-on labs), des stands animés par des acteurs du monde du

développement ou des sociétés aux profils variés faisant appel à ces technologies (Docker, RedHat, GitHub, Thales, Michelin et autres .. .. ..), sans oublier une petite soirée, afin de rendre l'événement plus festif et encourager à la rencontre, car n'oublions pas, Devoxx, c'est avant tout une histoire de rencontres !

C'est l'occasion idéale pour croiser d'anciens collègues ou même pour dénicher les projets qui seront nos futures missions.

Durant ces trois jours, on découvre forcément des outils intrigants et des technologies que l'on a immédiatement envie de tester sur nos projets.

Devoxx Paris, here we go !

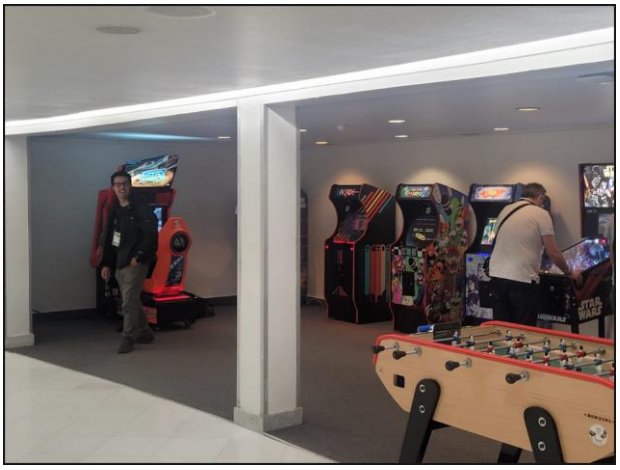
Le thème phare de l'édition de cette année était sans surprise l'IA, avec pas moins de 65 sessions dédiées. C'est un chiffre impressionnant, mais il y a tellement de conférences sur des thématiques variées qu'il est tout à fait possible d'esquiver le sujet si l'on fait une petite overdose.

Nous vous conseillons d'ailleurs de bien vous préparer si vous prévoyez de vous y rendre, car organiser son planning est un vrai défi, tant les sujets sont nombreux et intéressants.

Heureusement la plupart des conférences sont disponibles quelques semaines plus tard sur la chaîne YouTube de l'événement. On peut donc suivre ses thématiques favorites ou simplement se laisser porter par sa curiosité. Rien ne vous empêche de déambuler dans les couloirs, faire confiance à votre instinct, lire les panneaux informatifs devant les salles et décider d'entrer au dernier moment si le sujet vous interpelle.

Durant l'événement, parcourir les stands est non seulement l'occasion d'occuper le temps entre différentes sessions, mais aussi de faire le plein de goodies !

On a pu compléter nos collections de chaussettes et de porte-clés. Nous avons aussi tenté notre chance aux nombreuses tombolas, même si aucun de nous n'est reparti avec un Lego Star Wars. En revanche, la machine à pinces de GitHub n’a pas résisté à plusieurs tentatives, et nous avons bien profité des jeux d’arcade pour nous affronter entre deux conférences.



L'expérience continue également dans les couloirs, on profite du trajet vers la salle suivante pour discuter de ce qu'on vient d'apprendre.
Toutefois, nous vous conseillons de ne pas trop passer de temps dans les couloirs, certaines présentations sont tellement populaires que les files d'attente font plusieurs dizaines de mètres, sans garantie de pouvoir entrer.

Concernant les conférences, nous avons pu en voir un grand nombre dont les sujets étaient variés comme (liste non-exhaustive) :

- “Comment l’IA a transformé la création graphique de Devoxx France”, présenté par Nicolas Martignole et Stéphane Itschner, qui nous proposaient un retour plus créatif sur les coulisses visuelles de Devoxx France.

- “UDA — Unified Data Architecture à Netflix”, talk portant sur la manière dont Netflix cherche à unifier ses modèles métier autour d’un cadre conceptuel commun. L’enjeu est classique dans les grandes organisations : comment éviter que chaque équipe, chaque produit ou chaque domaine reconstruise sa propre vision des données, avec ses propres définitions, ses propres ambiguïtés et ses propres silos.

- “Stop à la dette documentaire : Industrialisez vos ADRs et READMEs avec Spec-kit” venant rappeler un sujet que l’on sous-estime souvent : la documentation aussi peut générer de la dette. Des ADRs absentes, des README obsolètes, des décisions non tracées… tout cela finit par ralentir les équipes. L’approche proposée était d’industrialiser davantage cette documentation, notamment via GitHub Copilot et Spec-kit, pour en faire un vrai support d’agilité plutôt qu’un cimetière de bonnes intentions.

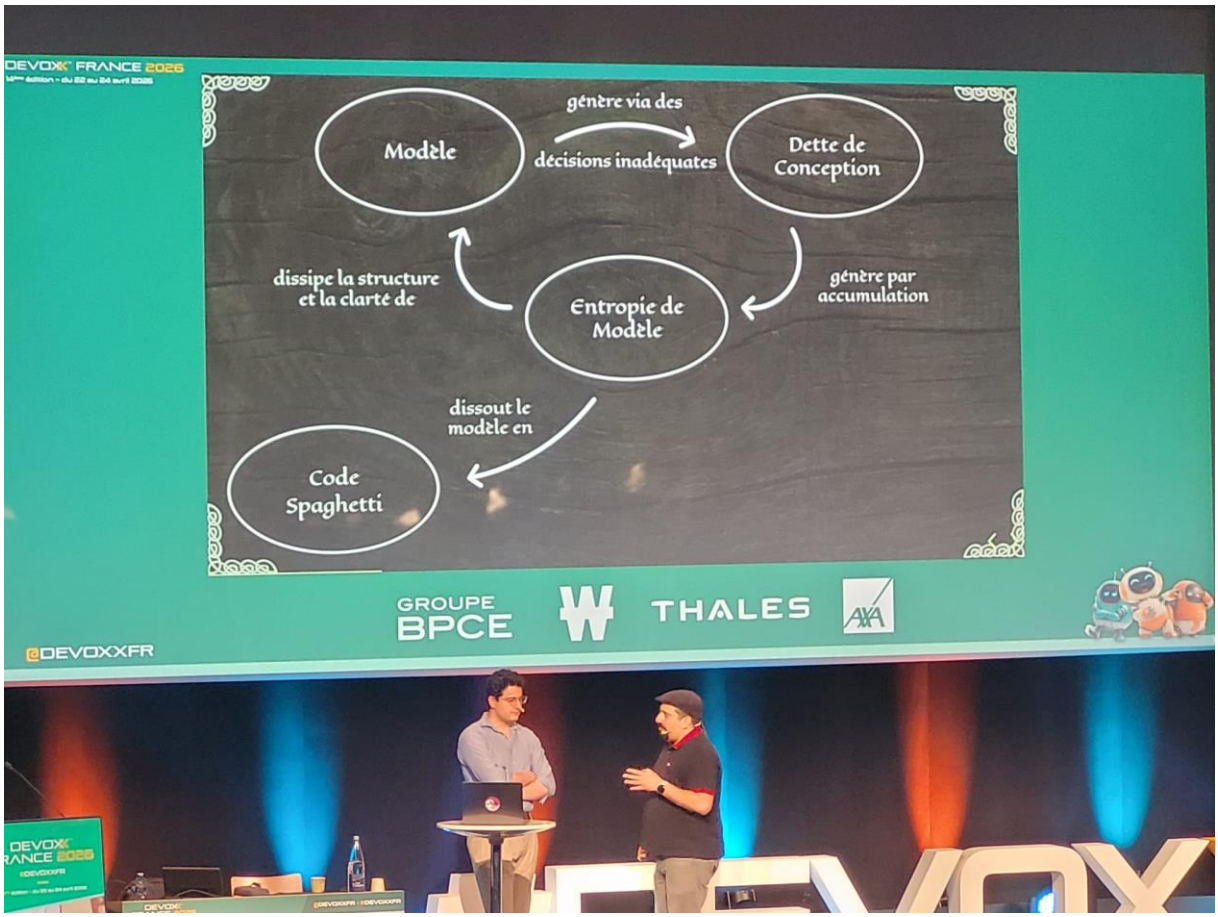
Nous vous proposons de vous détailler celles qui nous ont le plus intéressé :

- “Vous pensiez que la dette technique était la pire ? Voici la dette de conception”
- “Les design patterns agentiques dont vous êtes le héros”
- “L’éloge de la simplicité”
- “Se retourner la tête avec les quiz Java de José et Jean-Michel"

La dette de conception

L'idée centrale : parfois, ce que l'on appelle "dette technique" n'est que le symptôme d'un problème plus profond, lié à la conception même du système.

Une mauvaise découpe métier, des responsabilités floues, des abstractions mal placées ou une architecture qui ne reflète plus le domaine peuvent mener à des logiciels difficiles à faire évoluer, voire à des micro services qui ne sont finalement que des monolithes distribués.



C'est un sujet particulièrement intéressant pour nous, parce qu'il rappelle que la qualité logicielle ne se résume pas à "avoir du code propre".

On peut appliquer des bonnes pratiques, avoir des tests, du linting, des pipelines, et malgré tout construire un système bancal si le design initial ou l'évolution du modèle ne suit pas.

Les Designs Patterns Agentiques :

Ce talk permet de pouvoir répertorier les différents types de modèles agentiques (composé d'un ou plusieurs agents) ainsi que leur architecture.

On y retrouve plusieurs patterns comme :

- Retrieval Augmented Generation
    - Fourniture des données à l'agent afin de compléter son context.
- Progressive Disclosure (skill)
    - Guidelines avec fourniture d'outils pour limiter le contexte mais permettre un raisonnement plus précis et incrémental.
- Hierarchical Agent Decomposition
    - Découpage en plusieurs agents spécialisés pour effectuer une tâche.
- LLM-as-Judge
    - Analyse du travail d'agents par d'autres agents spécialisés.
- GOAP (Goal Oriented Action Planning)
    - Définition du but à atteindre au lieu de la fourniture du process.
- Ralph Wiggun & Feedback Loop
    - Boucle d'action permettant une résolution via plusieurs itérations.

Cette conférence permet de pouvoir connaître un ensemble d'architectures existantes, et aussi de se rendre compte que l'utilisation des agents est une pratique qui évolue constamment !

L’éloge de la simplicité :

Actuellement, les plannings prennent une part importante dans les organisations agiles, mais ces plannings sont paradoxalement un frein à l'agilité car :

- Les plannings sont souvent faits de manière rigide

- Ils sont rarement réalisés dans le pur intérêt du client (peu de clients consultent les roadmaps des produits)

- Ils sont la plupart du temps faux

Nous avons tous de nombreux exemples de planning qui sont souvent dépassés, mais pourquoi en est-on arrivé là ?

Pour plusieurs raisons :

- les plannings rassurent, ils ajoutent de la prédictibilité

- On ne sait pas faire autrement …

Frédéric LEGUEDOIS propose justement d'observer un type d'organisation simple, efficace (scientifiquement prouvé), agile (s’adaptant aux imprévus) et orienté 100% client :

L'organisation d'un service d'urgence !

Un service d’urgence médicale répond à plusieurs règles :

- Efficacité de l’organisation prouvée scientifiquement

- Priorisation selon l'urgence médicale (taux de mortalité)

- Communication constante entre les différents services (médecins, radiologues, etc.)

- Backlog simple et évolutif

Cette organisation est simple (exemple de la boite à chaussure contenant les cas classés par priorité), permet une adaptation constante, est au cœur des priorités.

Elle peut tout à fait convenir à des petites comme à des grandes organisations et répond bien plus à l'esprit "agile" que les plannings dans leur forme actuelle.

Cette présentation permet de montrer (avec humour, un format théâtre/stand-up et des preuves à l'appui) qu'il existe des alternatives aux plannings, qui nous enferment plus souvent dans un schéma prétendu prédictif, mais de prenant pas en compte les réalités du terrain, ainsi que la priorité: la résolution des problèmes du client.

Quizz Java (José PAUMARD, Jean-Michel DOUDOUX) :

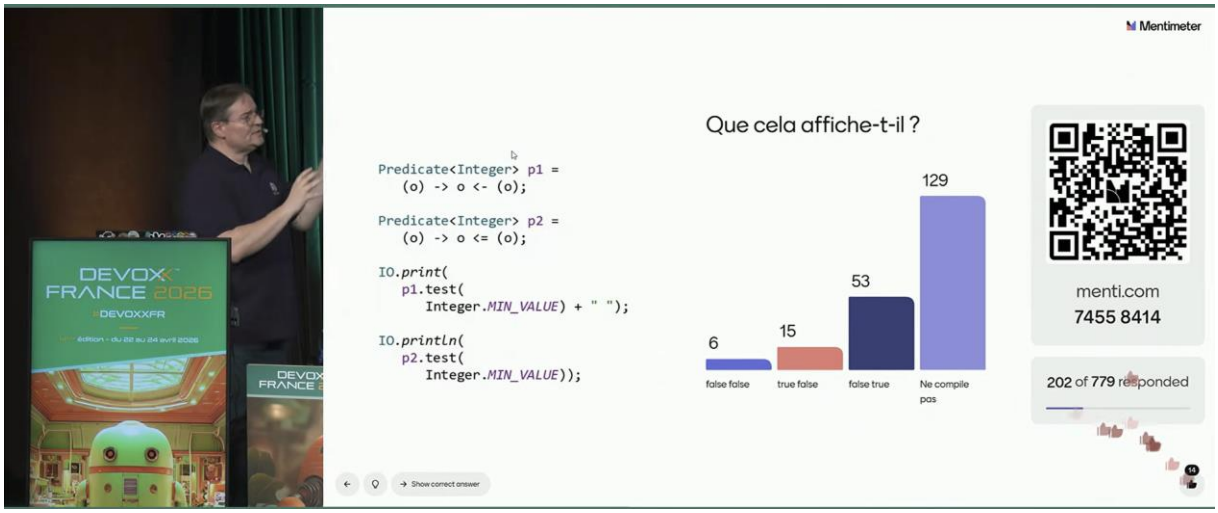
Le format de cette présentation était assez original, le but était un quiz en direct avec possibilité de voter via son smartphone, avec 10 questions sur JAVA (avec 4 réponses possibles).

Les questions portaient sur le résultat de l’exécution d’un code (ou sa non compilation/exécution), comprenant des instructions qui, à premier abord, sont simples.

Cependant (car il y a souvent un “cependant”) on constate justement qu’on ne connait jamais assez les fondamentaux, et qu’il est très facile de se tromper, ou encore de se faire influencer, car le nombre de votes par questions est visible durant l’épreuve !

José PAUMARD et Jean-Michel DOUDOUX nous montrent qu’il est facile de se faire piéger par des fondamentaux et qu’on peut toujours apprendre, même sur des bases !!

Prenons un exemple de question:



A votre avis, quel est le résultat?

Il est difficile de répondre à cette question si les Predicates vous sont inconnus.

Un Predicate est une interface fonctionnelle, ce qui permet d'executer une méthode "test" prenant un paramètre défini (Integer pour notre cas) dont on détaille l'implémentation, afin d'obtenir un résultat booléen (true ou false).

Dans ce cas précis, le piège se trouve dans la définition de la fonction:

Nous sommes en présence d'une expression lambda: (o) représente le paramètre de la fonction la flèche -> représente la séparation pour passer des arguments au corps de la fonction et o <- (o) le corps de la fonction.

On peut donc décomposer la déclaration de cette manière:

(o) -> o <- (o)

(o) -> (o < - (o))

le symbole < est en fait un "inférieur ou égal" et le - est en fait un negatif, qui, aussi surprenant que ce soit, n'a pas besoin d'être accolé au (o) pour vouloir dire -(o).
L'expression lambda donne donc l'impression d'une symétrie avec le o comme centre, et fait facilement perdre de vue que c'est bien une expression lambda.
Le resultat de cette méthode est donc "o inférieur à -o" ce qui donne false pour tout entier.

Une fois ce subterfuge découvert, il est facile de déchiffrer le symbole <= comme un "inférieur ou égal" pour le second prédicat, ce qui donne comme résultat "false true".

Nous vous laissons découvrir les autres questions dans la vidéo de cette présentation, afin que vous puissiez vous amuser et apprendre, autant que nous avons appris et nous sommes amusés.

Les Stands :

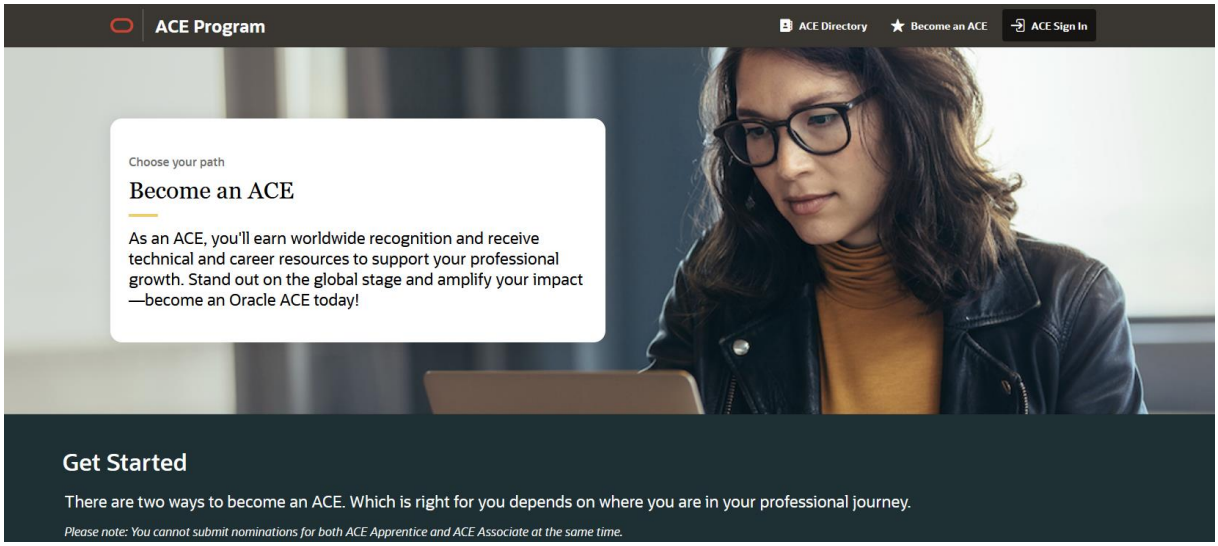
Durant ces 3 jours, nous avons pu également faire le tour de certains stands, qui proposaient non seulement des animations mais aussi des nouveautés :

Oracle Ace:
Oracle dispose d'un programme composé d'une communauté, regroupant des personnes qui connaissent et utilisent les produits oracle.

Les points forts de cette communauté sont :

- Documentation sur les différents produits Oracle comme Java, les bases de données, etc.)
- Entraide entre membres avec soutien de spécialiste

Petite information intéressante, 1 passage de certification gratuit/an pour les membres.



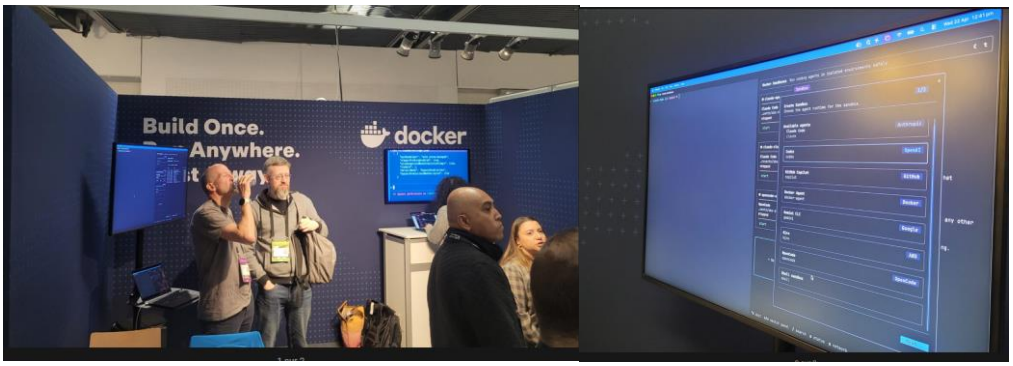
Docker Sandbox :

Docker propose un nouveau produit assez intéressant.

Afin de pouvoir rendre les outils du type Claude code, Codex, Kiro ou Open Claw plus efficaces, il est parfois nécessaire de leur laisser des droits possiblement impactants sur notre machine. Cependant, l'accès à certaines parties de l'ordinateur peut s'avérer fatale, comme une réécriture de fichiers système ou l'accès à des données confidentielles.

Docker a donc sorti des MicroVM servant de sandbox, permettant d'utiliser ces outils, tout en les isolant de la machine principale, permettant de limiter non seulement les droits mais aussi les défaillances possibles des agents IA.

Nous vous préparons bien sûr une petite présentation sur ce produit avec par exemple, l'implémentation de certains design pattern d'architecture d'agents !



Conclusion :

Cet événement fut vraiment une belle occasion de pouvoir non seulement se mettre à jour, mais aussi, de pouvoir découvrir de nouvelles pratiques, acquérir de nouvelles connaissances.

On aurait pu vous détailler encore beaucoup de choses, il y a eu beaucoup de conférences, beaucoup de visites de sites, mais il nous faudrait bien plus qu'un simple article pour le présenter.

Nous avons pu sortir de cette conférence avec beaucoup de connaissances, de nouvelles idées que nous allons vous partager au fil de l'eau, (articles, ateliers, etc.).

Encore merci à Objectware pour cette belle aventure et à très bientôt pour le partage, de manière plus détaillée, de nos différentes trouvailles.

Liens:

Chaîne youtube:
https://www.youtube.com/@DevoxxFRvideos/videos

Talks:

“Vous pensiez que la dette technique était la pire ? Voici la dette de conception” :

https://www.youtube.com/@DevoxxFRvideos/search?query=dette%20conception

“Les design patterns agentiques dont vous êtes le héros” :

https://www.youtube.com/watch?v=c092bWijLhU

“L’éloge de la simplicité” :

https://www.youtube.com/watch?v=7MpLQLKljG0&t=125s

“Se retourner la tête avec les quiz Java de José et Jean-Michel" :

https://www.youtube.com/watch?v=7bqt1Qp7Y2k