LEARN FAST TO BUILD FAST LE MONDE CASE STUDY

EUROPEAN LEAN IT SUMMIT - OCT. 2014

Ismaël Héry - Le Monde

ismael.hery@gmail.com - @Ismael_Hery

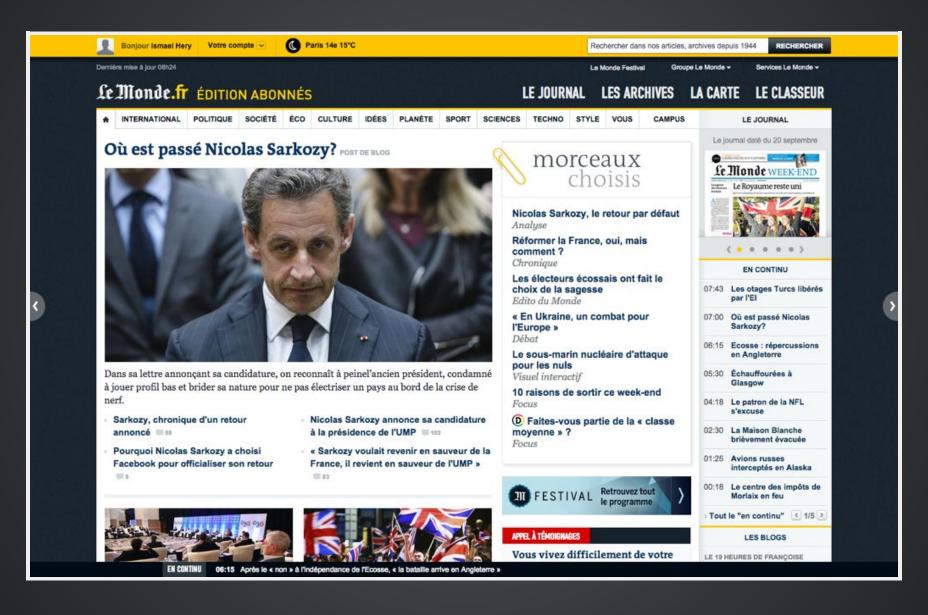
- Agile coach @OCTO until 2011
- Head of Software Development at le Monde.fr until 2013
- Delivery Manager for le Monde new print and digital content management plateform

WHAT WE DO AT LE MONDE

FREE WEB SITE (60 MILLIONS MONTHLY VISITORS)



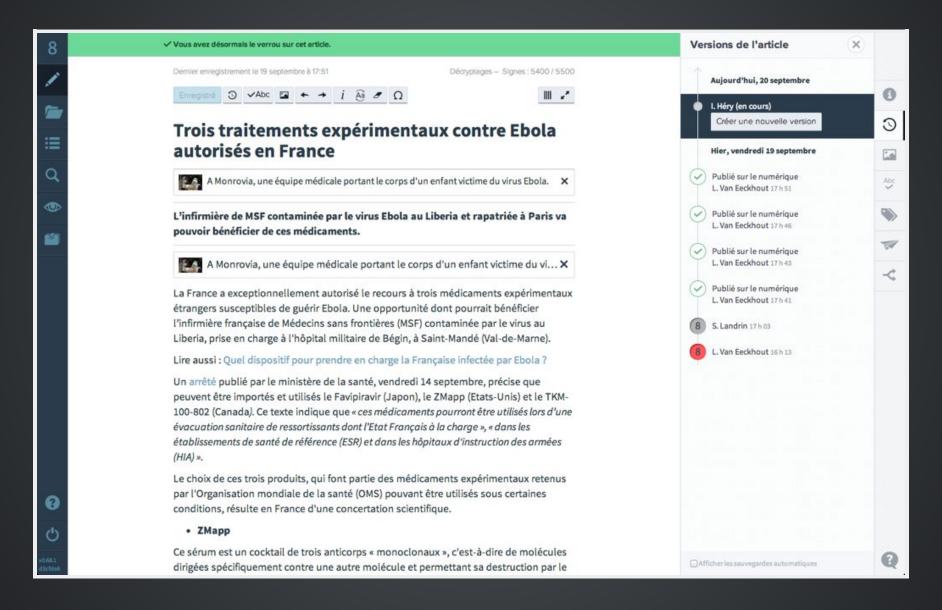
PREMIUM WEB & APPS (70 000 PAYING SUBSCRIBERS)



MOBILE APPS, IOS AND ANDROID



CMS FOR BOTH PRINT AND DIGITAL CONTENT



WHAT'S TYPICAL IN OUR CONTEXT

CLASH OF TWO CULTURES



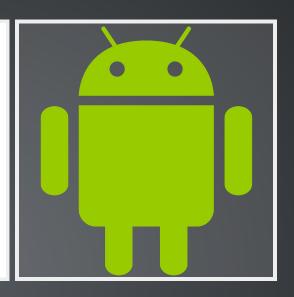


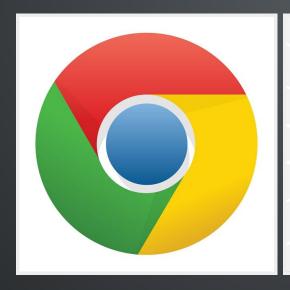
VERY HARD TIME ECONOMICALLY



YOUNG (& UNMATURE) TECHNOLOGIES





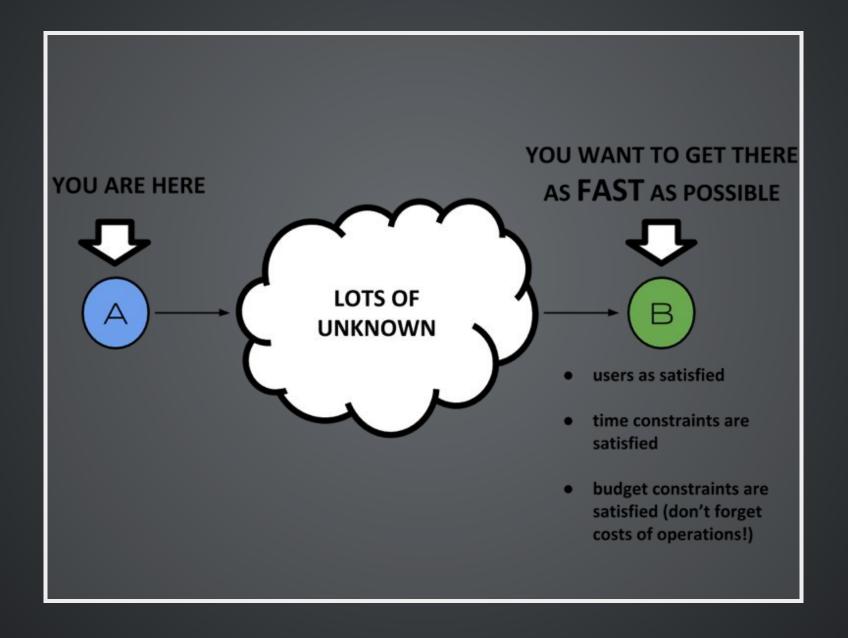




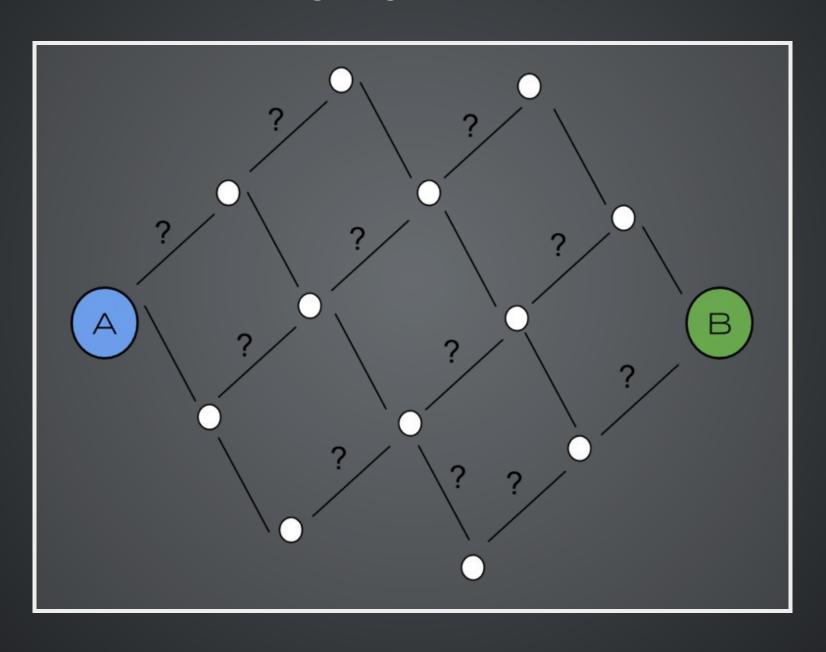
PROJECTS END ON TIME



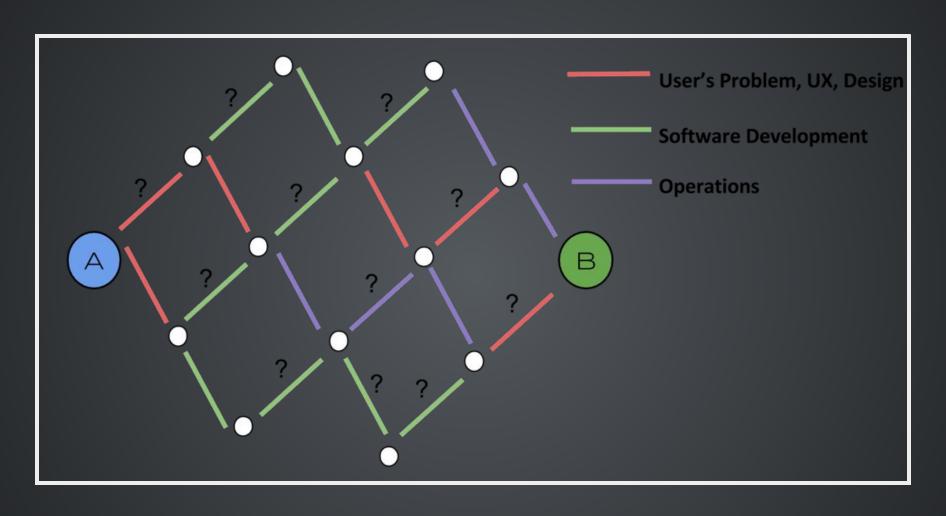
OUR PROBLEM



FASTEST PATH?



MAIN DOMAINS OF UNKNOWN



PRODUCT DEFINITION UNKNOWNS EXAMPLES

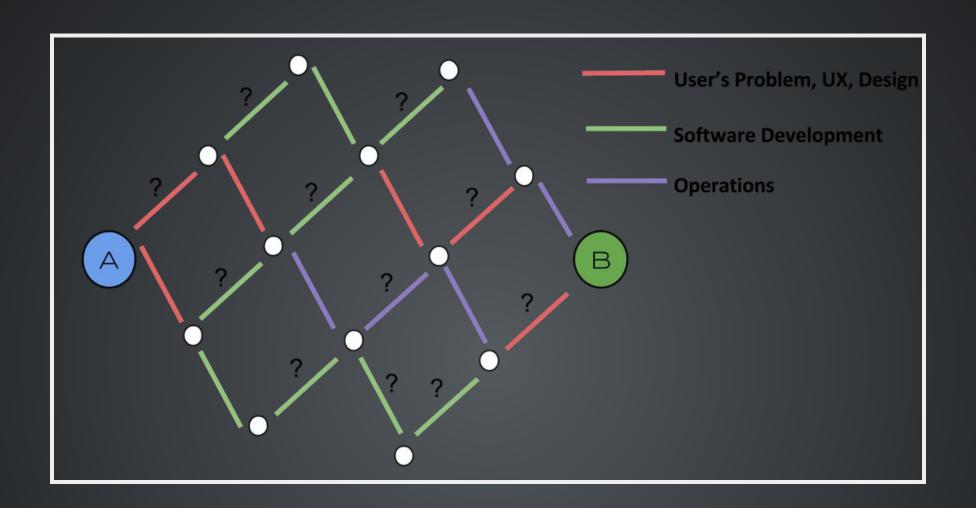
- Is this user problem a real issue or is it something *I* see as an issue?
- Will it work as expected once in the user's hands?
- How much should we invest in this feature compared to that one?

SOFTWARE DEVELOPMENT UNKNOWNS - EXAMPLES

- Should we use that framework or that one?
- How long does it take to develop that kind of feature?
- Is it possible to have automated tests with this new technology?
- Where will the next bottleneck appear when we get 10x more users?

OPERATIONS UNKNOWNS - EXAMPLES

- How long and how much effort will it take to deploy a new release after QA?
- What will be the main weaknesses of this new system once in production?
- What are most vital dashboards and alerts for this new system?
- How many more servers do we need to support the expected load?



OUR PROBLEM

TO KNOW IS TO KNOW THAT WE KNOW NOTHING

IF WE WANT TO BUILD FAST, WE NEED TO SOLVE A LOT OF UNKNOWNS FAST

HOW CAN WE AMPLIFY AND SPEED UP TEAM'S LEARNING?

OUR HYPOTHESIS

LEARN IN ALL CRITICAL DOMAINS, DON'T EXCEL AT ONE AT THE EXPENSE OF OTHERS

DON'T WAIT TO LEARN

PRACTICES WE USE TO SPEED UP LEARNING

USER TESTING, ASAP AND OFTEN





The how-to companion to the bestselling Con't Make Me Think!

A Common Sense Approach to Wild Lituability.

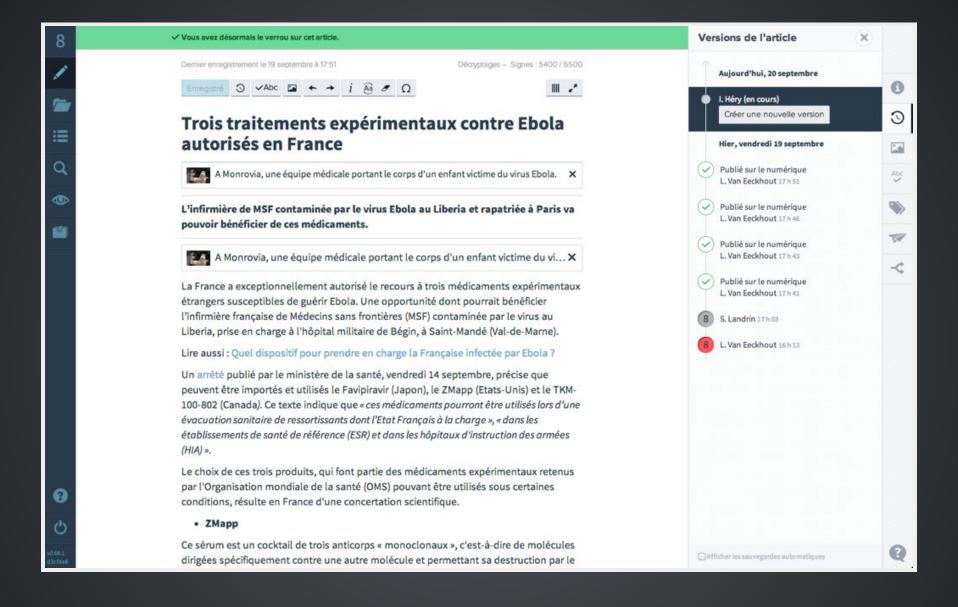


The Do-It-Yourself Guide to Finding and Fixing Usability Problems

Hypothèse à vérifier : L'utilisateur trouve immédiatement comment créer un article.			
Objectif atteint ?	□ Oui	□ Oui, avec mon aide	□ Non
Nombre de clics	1	□ entre 2 et 3	□ + de 3
Temps passé Indiquer éventuellement le temps passé ici :	□ rapide & intuitif	□ raisonnable mais pas intuitif (l'utilisateur a un peu cherché avant de trouver)	□ trop long
Nbre d'erreurs commises	□ aucune	1	□ + de 2
Rapidité de correction des erreurs	☐ Il corrige très facilement son erreur	□ II corrige difficilement son erreur	□ II corrige son erreur ave mon aide
	□ L'utilisateur semble satisfait	□ L'utilisateur ne semble pas satisfait	□ Difficile à dire ;-)
erreurs (Signes extérieurs de) satisfaction de l'utilisateur	erreur	erreur □ L'utilisateur ne semble pas	mon aide

USER TESTING STORIES FROM LE MONDE





DEPLOY (TOO) EARLY

FAKE A LOT

SUPPORT YOUR EARLY USERS

DEPLOY INVISIBLE STUFF IF YOU CAN'T DEPLOY VISIBLE STUFF

FAKE A LOT





TAKE GREAT CARE OF YOUR BETA TESTERS

- Recruit them wisely
- Encourage them
- Give them something back regularly

IF YOU CAN'T DEPLOY FOR YOUR USERS, AT LEAST DEPLOY FOR YOUR DEVS AND OPS

- Deploy sub parts technically testable in production
- Deploy early and progressively

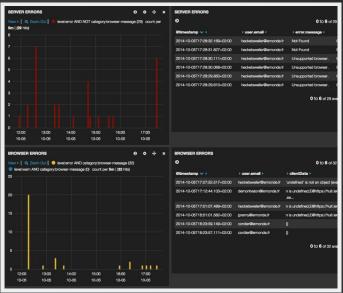
WHAT WE GET FROM DEPLOYING VERY EARLY

- Focuses product people on improvements really needed by your users
- Focuses ops on real production problems (not the one they wish they have)
- Improves system robustness very early
- Fuels continuous improvement based on concrete problems
- Creates a sense of urgency and gets more engagement

EARLY DEPLOYMENT STORIES FROM LE MONDE







WHEN IN DOUBT, CHOOSE THE PATH OF MAXIMIZED LEARNING

WHEN FACING A CHOICE BETWEEN TWO SOLUTIONS WITH GROSSLY THE SAME ROI, PREFER THE ONE THAT TEACHES YOU THE MOST!



WHAT WE HAVE LEARNED

- It rarely works as expected, even when we do our homework correctly (design studio, UX tests etc).
- Deployment and use in production generates the biggest amount of knowledge (by an order of magnitude).
- Still lots of improvements ahead at doing user testing.

CONCLUSION

- Don't excel at learning in one domain at the expense of the others (Theory of Constraints applied to learning?).
- Favor tools and practices that amplify both execution AND learning.

THANK YOU! Q & R

@Ismael_Hery