

MariaRosa Quintero Bernabeu

JOURNÉE
FRANÇAISE
DES TESTS
LOGICIELS

Le Risk-based Testing par couches :
le principe des lasagnes

BEFFROI DE MONTROUGE





Speaker

MariaRosa Quintero Bernabeu
Manager assurance qualité
Mariarosa.Quintero.Bernabeu@
mgm-tp.com

- 1999 Licence de Chimie
- 2007 Doctorat de Biochimie
- 2008-2009 Chercheuse en Biochimie
- 2010-2018 communication scientifique
- Depuis 2019 – chez mgm technology partners France
- 2019 – 2020: QA pour un projet e-commerce pour un client grande distribution
- Depuis 2021: QA dans des projets pour le secteur public allemand
- 2023 ISTQB Certification Test Manager

Le Risk-based Testing par couches

Sommaire

Définition du problème : les risques dans le développement logiciel

Qu'est-ce qu'un risque ?
Pourquoi avons-nous besoin du Risk-based Testing ?
Les avantages du Risk-based Testing

La solution standard : l'approche RBT classique

Une approche scientifique : le modèle de prédiction Bayésien

Une solution avancée: le Risk-based Testing par couches : le principe des lasagnes

Logique Métier
Historique des tests
Release Scope ou Périmètre de la version
Allocation des testeurs
Couverture du code
Agrégation : la Heatmap ou carte thermique

Exemples d'utilisation

Logique métier - Évaluation manuelle ou Automatique
Historique des Tests
Ordre des tests basé sur les risques
Rapports sur les risques

Conclusion : Comparaison et avantages

Bénéfices pour la QA
Bénéfices pour l'entreprise

Définition du problème

Les risques dans le développement logiciel

Qu'est-ce qu'un risque ?

Risque = possibilité qu'un évènement impacte de façon positive ou négative

- *Le succès*
- *Le planning*
- *Le budget*
- *La qualité*
- *Les objectifs*
- *La réputation*

D'un projet



Lié au produit

Perte de fonctionnalité
Failles de sécurité



Lié au processus

Manque d'expérience
Retard



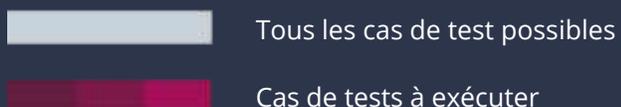
Externes

Exigences réglementaires
Évolution du marché

Pourquoi avons-nous besoin du Risk-based Testing ?

- Chaque projet a des ressources limitées : temps, budget, personnel...
- Tester et même juste créer tous les cas de test est **impossible**. Par exemple, une page de login simple pourrait générer des milliers de cas de test
- **Prioriser** est toujours nécessaire en QA

→ Tester d'abord les parties les plus critiques du projet



Les avantages du Risk-based Testing

Détection rapide des défauts les plus critiques



Augmentation de l'efficacité des tests



Plus grande satisfaction des clients



Diminution des coûts

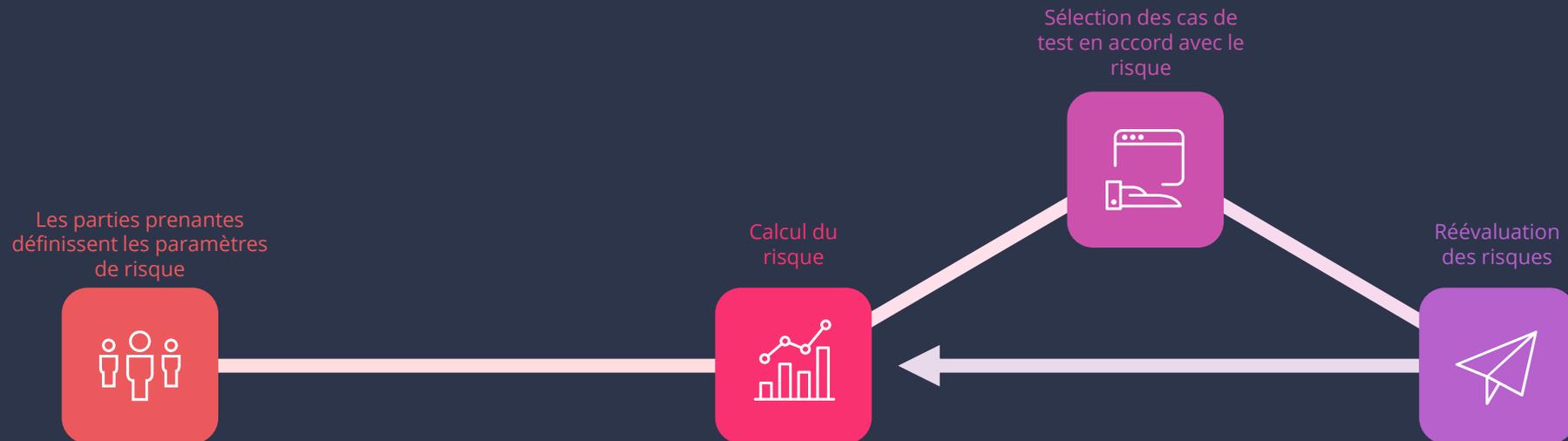


La solution standard
L'approche RBT classique

Solution standard

L'approche RBT classique

- Planification des activités de test basée sur les risques identifiés
- Objectif : maximiser l'efficacité des tests
- Différentes méthodes pour évaluer les risques
→ le plus souvent par l'estimation manuelle de la **probabilité** et de **l'impact**



Solution standard

L'approche RBT classique - Conclusion

AVANTAGES

Processus d'assignation des risques aux tâches de tests facile à comprendre
Priorisation des tâches selon le risque déterminé

INCONVÉNIENTS

- **Subjective** : l'évaluation peut varier en fonction de l'expérience de la personne
- **Manuelle** : l'évaluation de chaque tâche est fastidieuse et prend du temps
- **Non-évolutive** : une réévaluation constante est nécessaire
- Souvent la classification obtenue est très large

Une approche scientifique

Le modèle de prédiction Bayésien

Le modèle de prédiction Bayésien



Dr. Martin Varendorff, 2015

“Dans l'industrie, les tests sont généralement réalisés sous forte pression en raison de ressources limitées. Par conséquent, les tests basés sur les risques, qui utilisent les risques prédits pour guider le processus de test, sont utilisés pour sélectionner les cas de test.



A Bayesian Prediction Model for Risk-Based Test Selection

Hans-Martin Adorf¹, Michael Felderer¹, Martin Varendorff¹, Rolf Breu¹
¹agap technology partners, Munich, Germany
Email: {hans-martin.adorf, michael.felderer, martin.varendorff}@agap-gp.com
²University of Innsbruck, Innsbruck, Austria
Email: {michael.felderer, rolf.breu}@uibk.ac.at

Abstract—In industry, testing is commonly performed under severe pressure due to limited resources. Therefore, risk-based testing, which uses predicted risks to guide the test process, is employed to select test cases. To this end, risks have so far mainly been estimated ad hoc, but not systematically predicted on the basis of the defect history and defect costs. In this paper, we present a novel approach to risk-based test selection, which employs a comprehensive and versatile three-risk model taking defect probabilities and cost into account. It enables the prediction of a risk decrease that would potentially be used for test selection. We first define a generic Bayes risk decision criterion for test selection, and then implement and evaluate it in an industrial software development project, where it is intended to support decisions regarding the quality assurance process.

Index Terms—Risk Analysis, Bayesian Modeling, Prediction, Risk-Based Testing, Test Management, Test Selection.

I. INTRODUCTION

Risk-based testing (RBT) is a methodology used in software quality assurance (QA) for solving decision problems, particularly those related to selecting and prioritizing test cases [1], [2]. For some time now, these RBT approaches have gained great interest [3], but, as shown in a recent study [7], up to now they are mostly performed in an ad hoc fashion, e.g. not taking into account the defect history of the software under test (SUT). This inadequate procedure often leads to suboptimal decisions, which could considerably be improved upon by analytical decision support.

In this paper we present a novel approach to risk-based test selection. We build a Bayesian statistical model that allows us to predict the risk decrease when the associated cost could potentially be saved, if the test case were executed. Bayesian statistical models have successfully been applied before to predict effort [11], [27] and defects [21], [25]. But their potential for risk-based test selection and prioritization remains largely unexploited [19]. We introduce a statistical model describing a Bayes risk (BR) decision criterion [16]. We discuss the implementation of a BR predictor, and evaluate its performance in the context of a real-world software development project.

When compared to other valuable decision criteria such as maximum likelihood, maximum a-posteriori, or Neyman-Pearson, the BR decision criterion turns out to be more versatile, since, in addition to likelihood functions and prior probabilities, it may also take into account the crucial costs associated with making a correct or erroneous decision [15]. This renders the BR decision criterion highly relevant in any business environment.

II. INDUSTRIAL CONTEXT

The SUT of interest here consists of a custom, component-wise library, which is integrated into many commercial software products built for the German market. These products are installed in a very diverse set of environments, including all major client platforms for home users as well as enterprise operating systems. Additionally, the SUT is embedded in various runtime environments of major current programming languages. Not surprisingly the customer demands consistently high quality standards.

The SUT is released twice a year. Within each release cycle, eight to ten snapshots of the SUT are delivered to the QA team. An individual test run for a snapshot usually spans a week. The sequence of snapshots can be partitioned into three major groups. The first group of four to five snapshots marks the system development phase P_1 . QA tasks executed within P_1 are targeting defects from recent development activities. These defects are usually fixed right away, in order to reduce the risk of a defect pile-up. A second group of about three snapshots mark the adaptation phase P_2 , where the development team concentrates on fixing defects and on finishing late change requests. Tasks executed within P_2 are targeting all “releaser” defects, in order to avoid patch releases. The one or two snapshots in the third finalisation phase P_3 are release candidates. The tasks executed within P_3 verify that the latest corrections to the SUT did not introduce new defects.

The QA tasks to be executed within a test run are chosen from a fixed list. A task may consist of starting an automated test and evaluating the result, or executing a semi-automated test, or performing a purely manual job such as checking the documentation. The average effort for a task ranges from half an hour up to several days, and usually there is not enough time within a test run to execute all tasks.

At the beginning of each test run a test manager reviews, based on this review he assigns the tasks to be executed, trying to balance the gain from finding defects early with the effort spent. Obviously, the decisions of the test manager on how to select the tasks for a test run are critical. The goal of the presented approach is to assist the test manager in these decisions.

On purpose, the test manager’s role has been rotating within the QA team. Thus, each team member has considerable experience about testing the SUT. It is therefore natural to consider these well-trained sources, i.e. in our case, an experts, whose

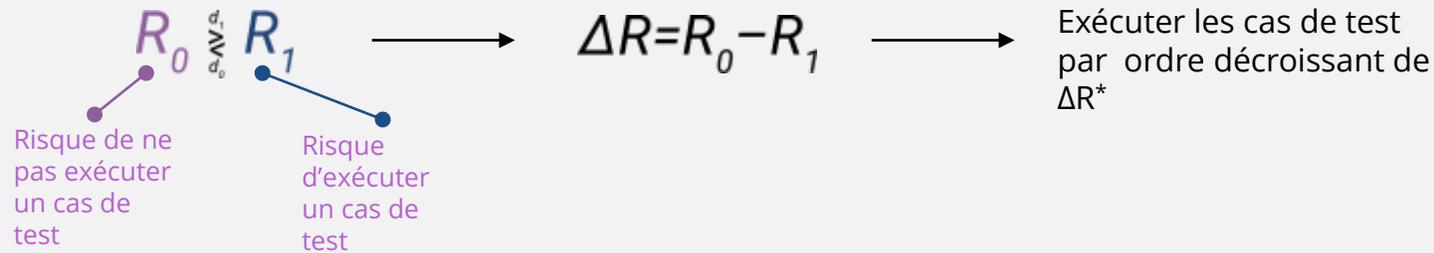
... is an attempt to provide and select test cases for test... by process, people and product factors, the test engineer service reduction. You and Choi [29] propose a test case... of components and other artifacts. This information is further

... which is totally independent of the data vector x . Here, make decision d_1 . Using equation (1), which already includes $P(d_{10})$ and $P(d_{11})$ are the prior probabilities discussed below. the risk-minimization, we can compare d_1 and d_2 .

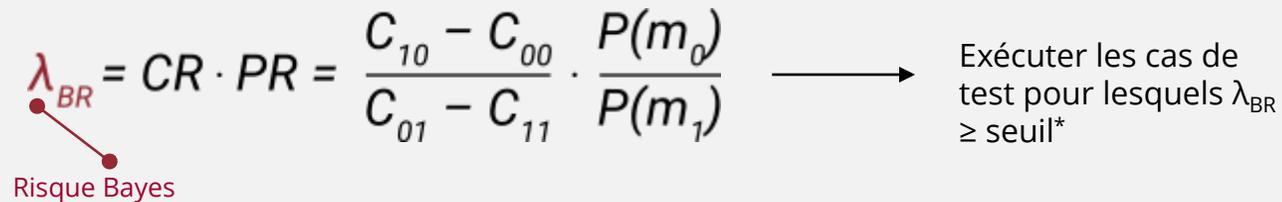
Adorf, H.-M., Felderer, M., Varendorff, M., & Breu, R. (2015). A Bayesian Prediction Model for Risk-Based Test Selection. In 2015 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 323-326). IEEE. doi:10.1109/SEAA.2015.37

Le modèle de prédiction Bayésien

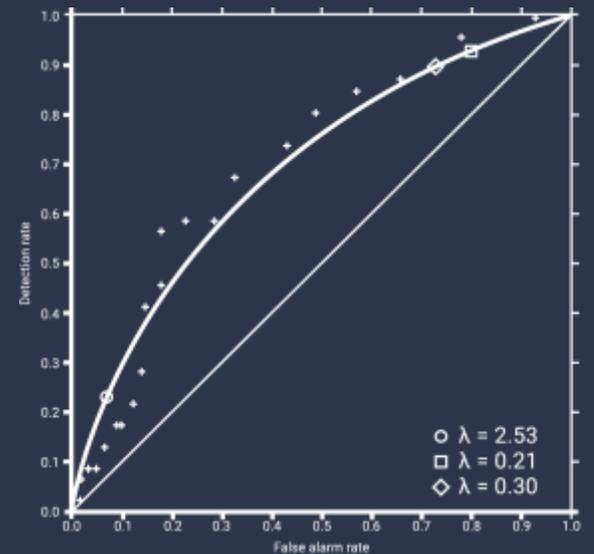
Budget non maîtrisé



Budget maîtrisé



Caractéristiques de fonctionnement du modèle de prédiction bayésien



- Empirical operating characteristics
- / Identify function
- ⌒ Theoretical operating characteristics
- ◇ Operating points corresponding to the calculated λ_{BR} in different project phases

* Adorf, H.-M., Felderer, M., Varendorff, M., & Breu, R. (2015). A Bayesian Prediction Model for Risk-Based Test Selection. In 2015 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 323-326). IEEE. doi:10.1109/SEAA.2015.37

Le modèle de prédiction Bayésien- Conclusion

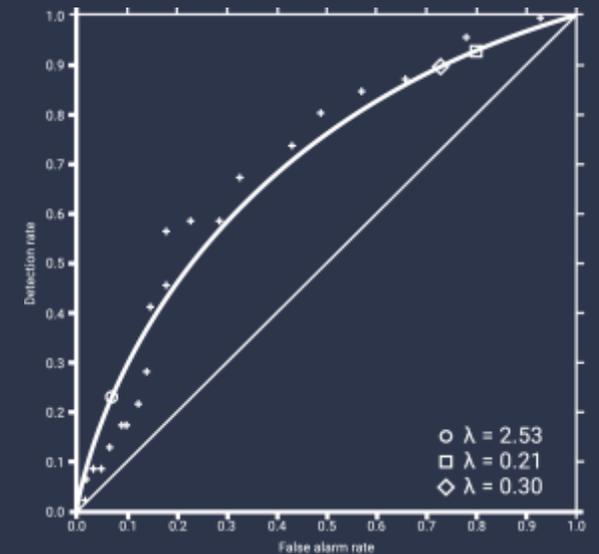
AVANTAGES

Identification fiable des cas de test à risque
Priorisation des tâches en fonction d'une réduction des risques quantifiable
Adaptabilité aux différentes phases de projet et de test
Normalisation des avis d'experts (non illustré)

INCONVÉNIENTS

Estimations manuelles nécessaires
Réévaluation constante nécessaire à chaque version
 Les avis d'experts sont des **estimations**

Caractéristiques opérationnelles du modèle de prédiction bayésien

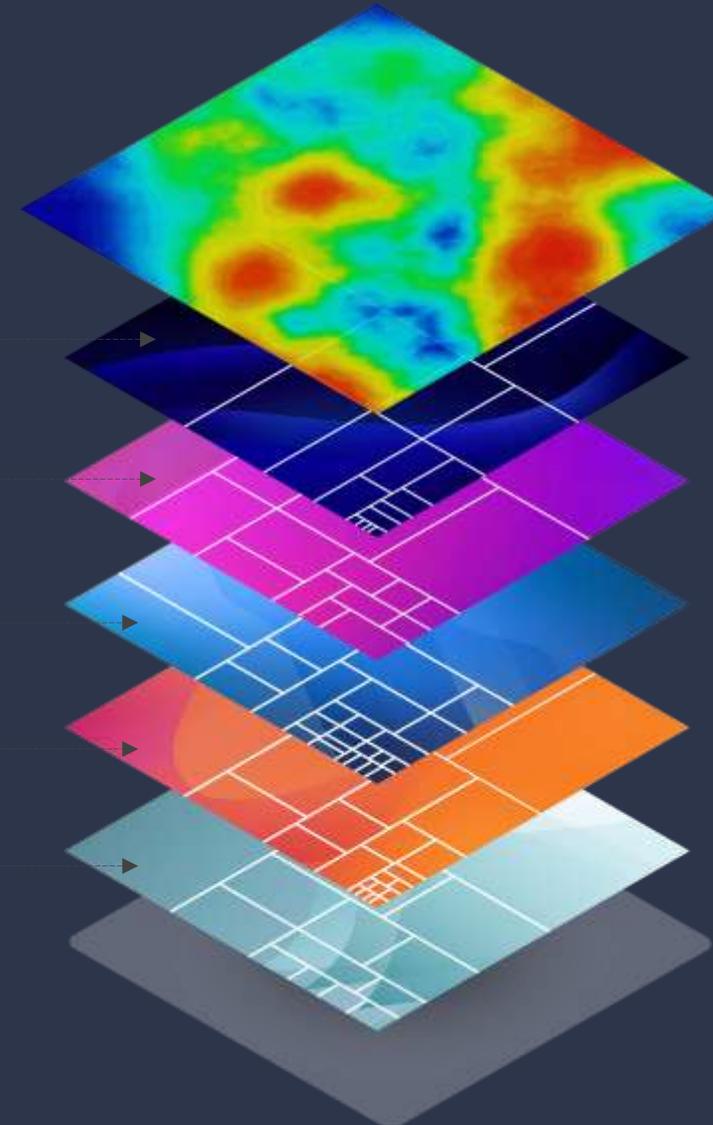


- + Empirical operating characteristics
- / Identify function
- Theoretical operating characteristics
- □ ◇ Operating points corresponding to the calculated λ_{BR} in different project phases

* Adorf, H.-M., Felderer, M., Varendorff, M., & Breu, R. (2015). A Bayesian Prediction Model for Risk-Based Test Selection. In 2015 41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 323-326). IEEE. doi:10.1109/SEAA.2015.37

Une solution avancée L'approche mgm: le Risk-based Testing par couches

Le Risk-based Testing par couches : le principe des lasagnes



Risk Heatmap
La carte thermique permet d'identifier les points chauds pour en dériver la priorisation et la planification.

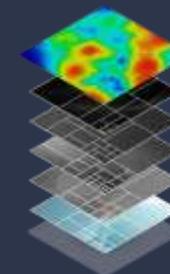
Solution avancée

Couches RBT – Logique Métier

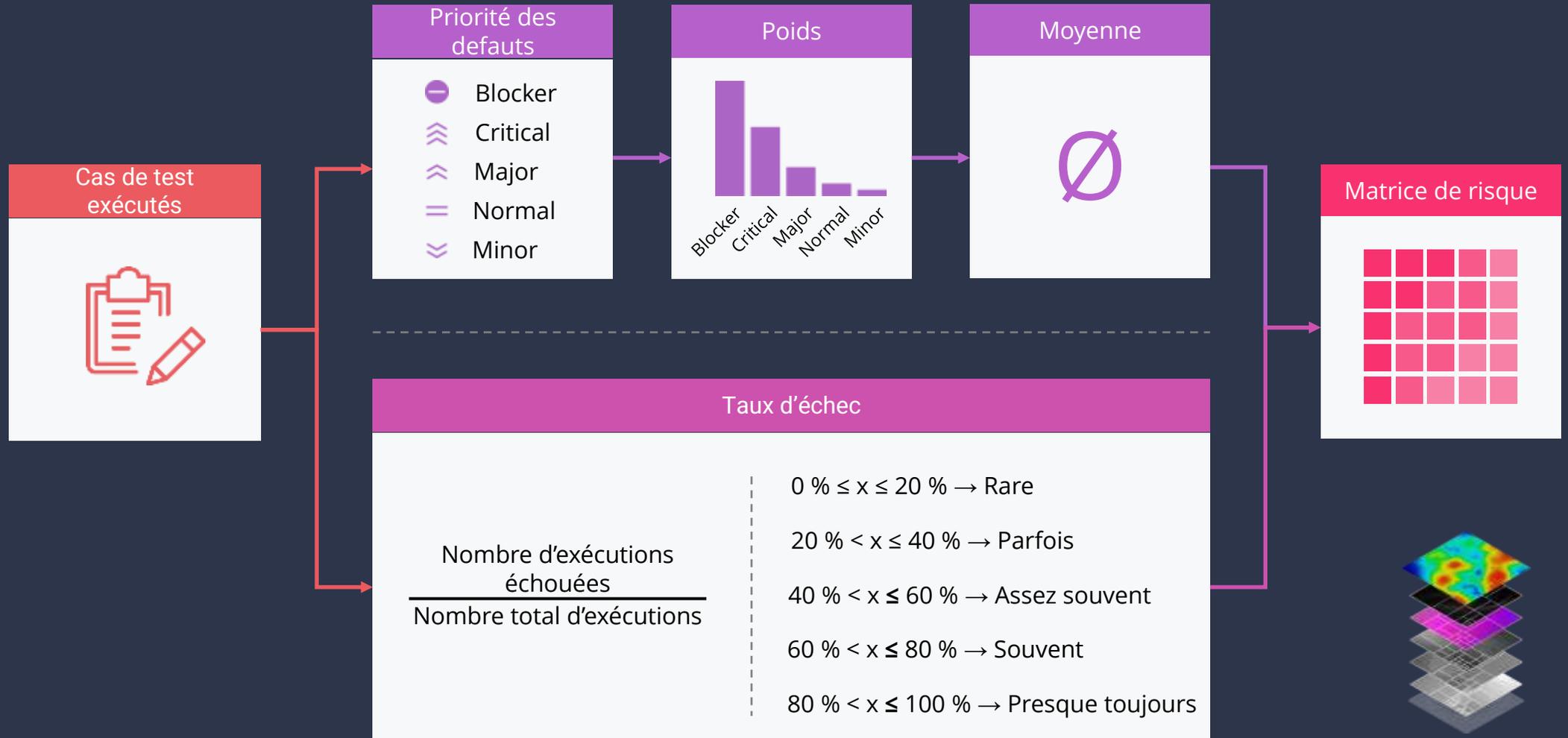
← Impact

Probabilité ↑	Impact				
	Grave	Majeur	Moderé	Mineur	Insignifiant
Très probable	-	5	3	-	6
Probable	2	14	21	24	38
Possible	-	8	13	104	59
Peu probable	1	10	34	58	84
Rare	3	7	20	42	43

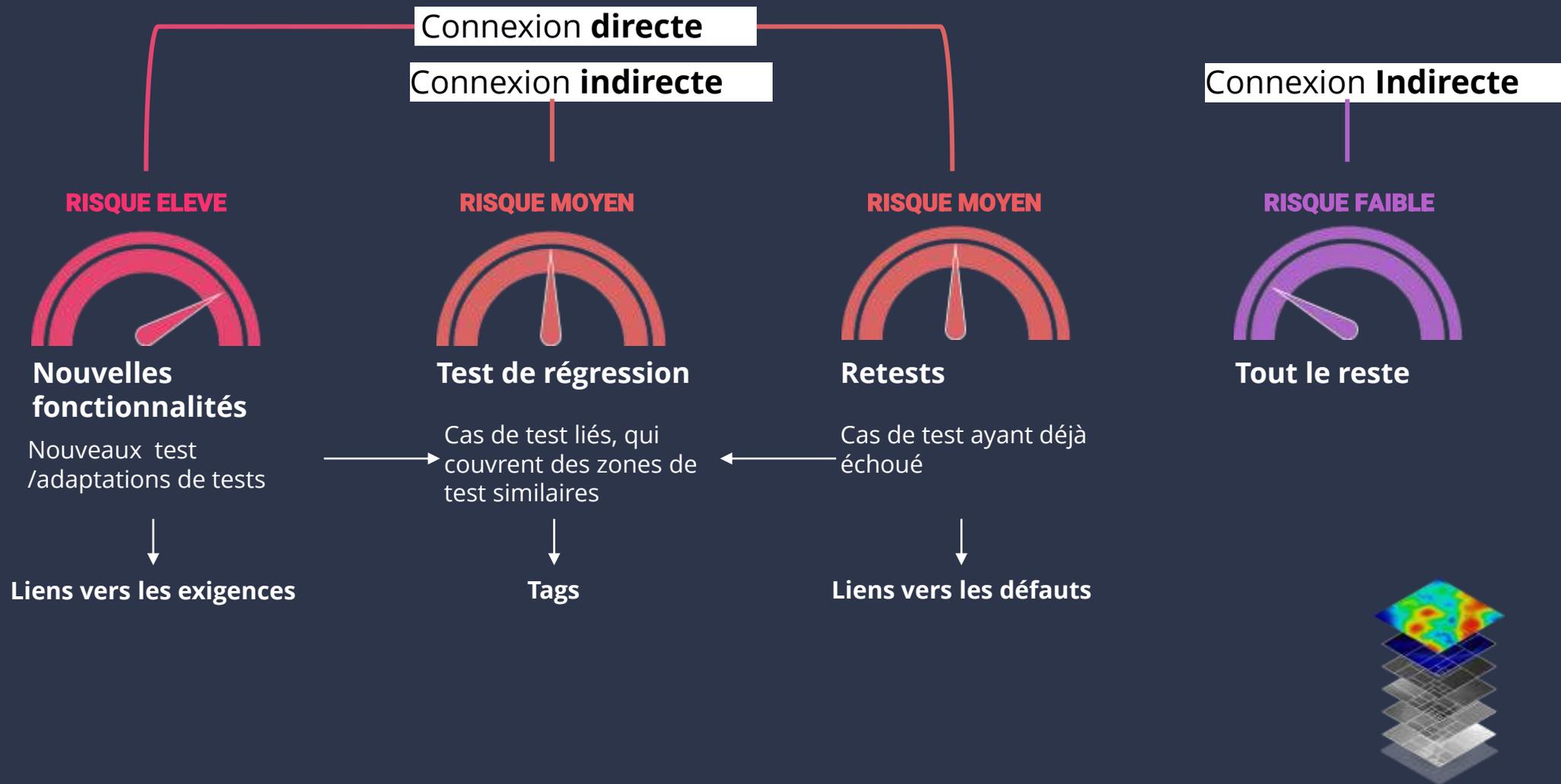
- Estimation manuelle de la **probabilité** et de l'**impact** → possibilité d'évaluation **automatisée**, basée sur la complexité et l'analyse de mots clés
- Attribution de valeurs de risque (nombres de Fibonacci) et dérivation de groupes de risque dans une matrice



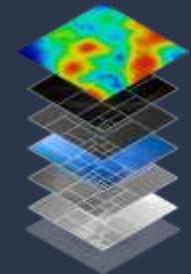
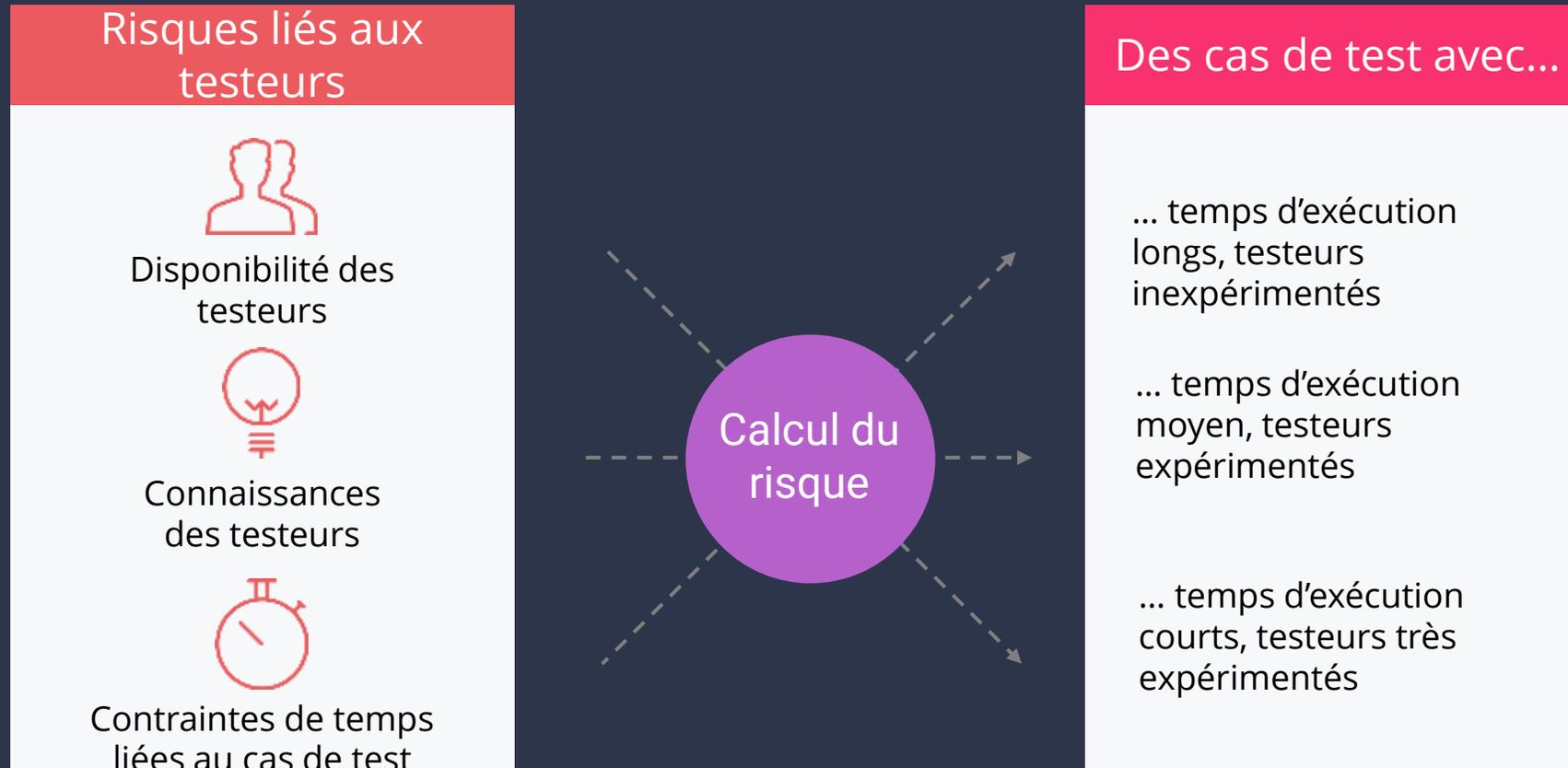
Couches RBT – Historique des tests



Couches RBT – Release Scope ou Périmètre de la version

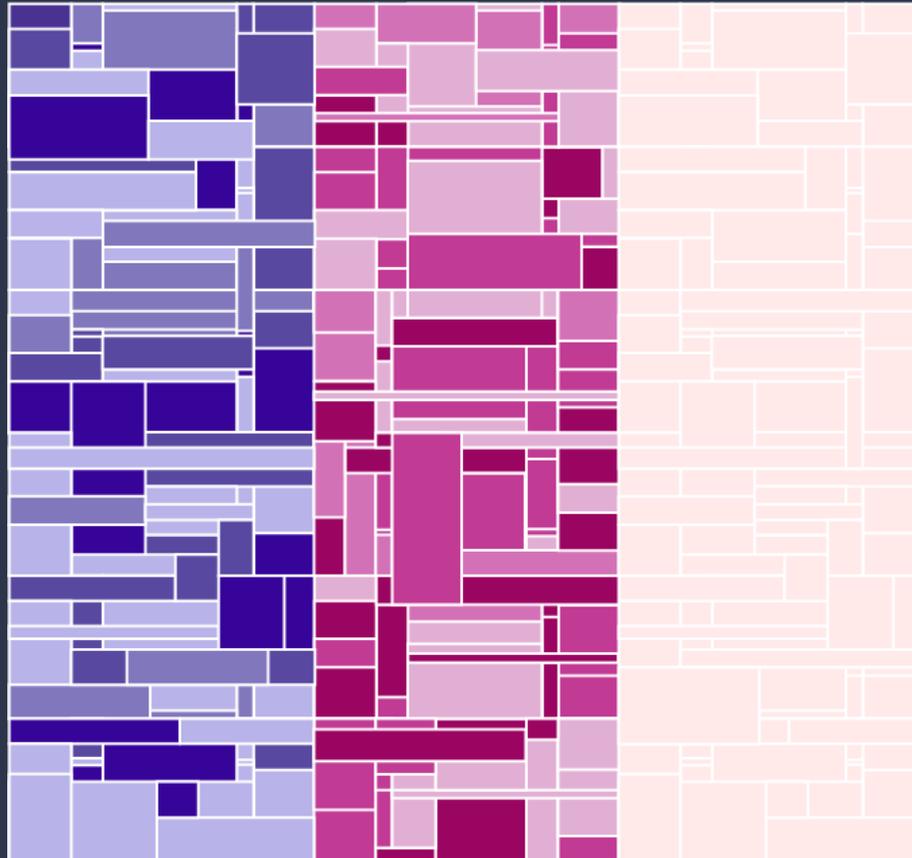


Couches RBT – Allocation des testeurs

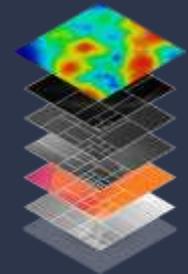


Solution avancée

Couches RBT – Couverture du code



- Toutes les modifications du code ne sont pas documentées ou évidentes pour les testeurs.
- Analyser le code (non) exécuté par les tests et identifier les potentiels risques
- Attribuer les risques en fonction des changements dans la base de code :
 - Nouvelles méthodes
 - Méthodes modifiées
 - Méthodes inchangées



Couches RBT – Agrégation : la Heatmap ou carte thermique

Idée

Estimation du **risque global** prenant en compte toutes les couches RBT
Reflète le **risque combiné** en tenant compte de toutes les perspectives

Méthode initiale : approche du « risque le plus élevé »

Le risque le plus élevé de chaque couche détermine le risque global

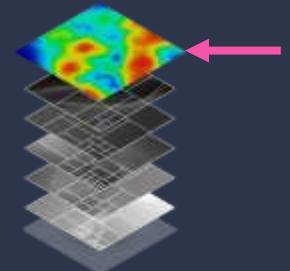
- +Simple, facile à établir
- Relativement rigoureux, sans nuance, quantification précise

Méthode affinée : Approche « numérique »

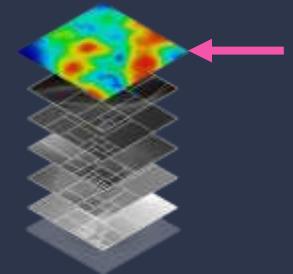
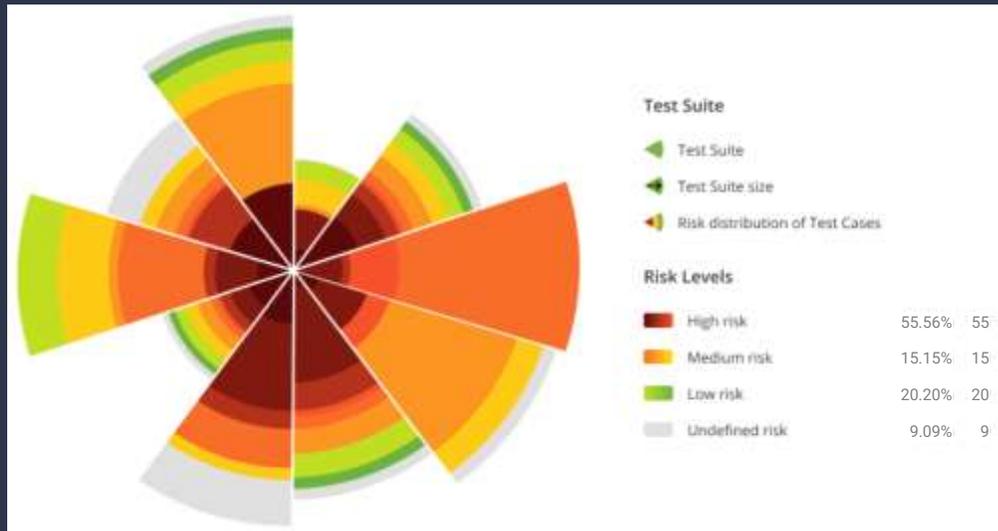
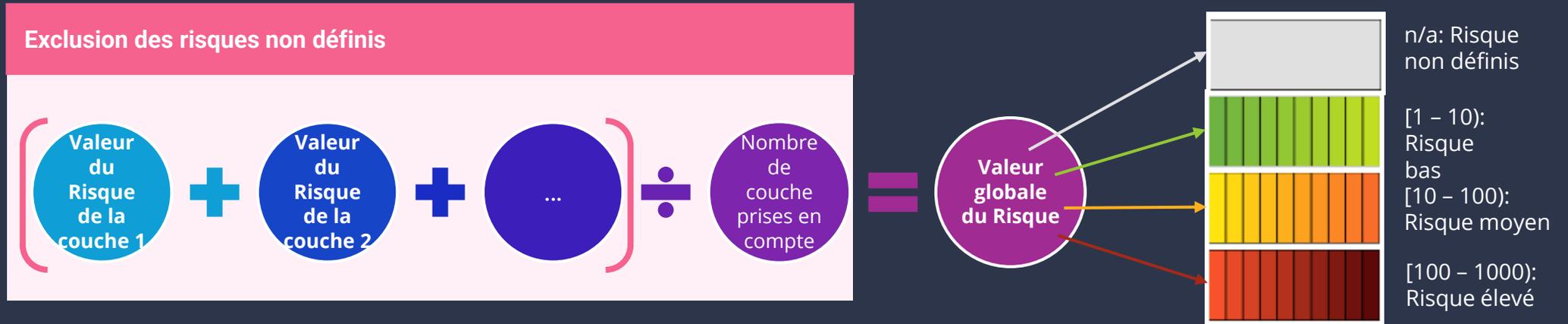
Attribuer des **valeurs numériques** de la suite de **Fibonacci** à chaque combinaison de paramètres par couche de risque
Former la **moyenne arithmétique** de chaque couche pour obtenir une valeur de risque globale

- +Plus granulaire et nuancée, flexible, pondérée intrinsèquement
- Nécessite davantage d'information

Low risk	Medium risk	High risk
1	13	144
2	21	233
3	34	377
5	55	610
8	89	987



Couches RBT – Agrégation : la Heatmap ou carte thermique



Exemples d'utilisation

Application pratique en assurance qualité

Logique métier - Évaluation manuelle

Saisies requises

Par cas de test

Paramètre de probabilité

Paramètre d'impact

Likelihood

- ▲ Almost Certain
- ▲ Likely
- ▬ Possible
- ▼ Unlikely
- ▼ Rare

Impact

- ▲ Serious
- ▲ Major
- ▬ Moderate
- ▼ Minor
- ▼ Insignificant

Calcul automatique

Par cas de test

Valeur du risque de la logique métier

Groupe de risques de la logique métier

Risk Level
↑ High (233)

Risk Level
↔ Medium (34)

Risk Level
↓ Low (1)

Risk Level
◇ Undefined

Logique métier - Évaluation Automatique

Saisies requises

Par cas de test: Activer/Désactiver le paramètre

Likelihood
Manual Auto

Almost Certain

Impact
Manual Auto

Minor

Par projet : Liste de mots-clés

Business Logic - Keywords on Test Case Impact

Score	Keywords
10	api, login
9	payment, credit card, price, discount
8	performance, checkout, order
7	gdpr, policy
6	role, permission, email, e-mail, accessibility, create
5	reflector, calculate, rules, validate, validation, search
4	select, edit, update, save, store, priority, customer
3	- no data -
2	export, delete, move, copy
1	- no data -

Données utilisées

Par cas de test
Occurrence des mots-clés
Complexité

Par projet
Complexité médiane du cas de test

Preconditions

The user is on the checkout page and has valid credit card information.

Test steps

No.	Action	Expected result
1	Enter shipping information. Select "Credit Card" as the payment method.	A field to provide credit card information ("Card Number" and "Cardholder") is displayed.
2	Click on the "Complete Purchase" button.	The user is redirected to the external payment page of the payment provider.
3	On the external payment page, double check card number and holder information, provide the CVV, and follow the 3D secure process (2 factor).	The process could be finished.
4	Once the process is finished and the payment reservation is successful at the payment provider, the user is redirected to the shop.	The confirmation page is displayed, including a success message.
5	Check mailbox.	A confirmation email has been sent to the user.

Calcul automatique

Par cas de test
Impact et probabilité
Valeur du risque lié à la logique métier
Groupe de risques liés à la logique métier

Likelihood

Possible

Impact

Serious

Risk Level

High (233)

Exemples d'utilisation

Historique des Tests

Related Test Runs		
Title	Test Case result	Test Case defects
Saint Nicholas Day	Passed	
Advent Sunday Dinner	Failed (new)	TMT-4675

$$\frac{\text{Nombre d'exécutions échouées}}{\text{Nombre total d'exécutions}} = 1 / 2 = 50 \%$$

Failure Frequency
= Sometimes

Test History
Risk
↔ Medium (89)

Priorité du ticket : critique

Pondération : 5

Moyenne : 5 / 1 = 5

Average Severity
▲ Intense

	Crucial	Intense	Substantial	Significant	Mild
Almost always	987	610	233	144	13
Often	610	377	89	55	5
Sometimes	377	89	55	21	3
Infrequent	144	34	21	5	2
Rare	55	8	5	3	1

Données masquées d'un système en production.

Ordre des tests basé sur les risques

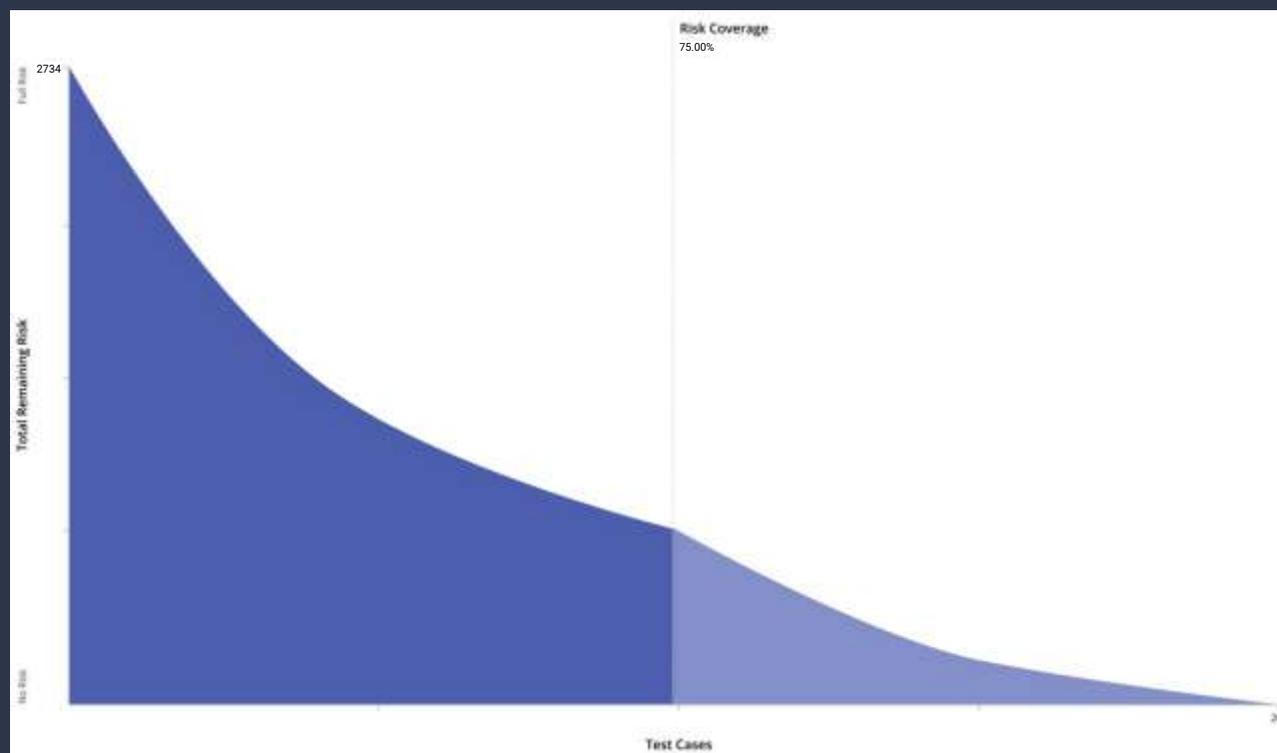
Title	ID	Result	Overall Risk	Risk Business Logic	Risk Test History	Risk Release Scope
Successful User Registration	C_1	Untested	High (658)	High (610)	High (987)	High (377)
Complete Purchase with Invoice	C_34	Untested	High (377)	High (377)	Undefined	Undefined
Login with Correct Credentials	C_3	Untested	High (252)	High (377)	Low (1)	High (377)
Complete Purchase with Credit Card	C_17	Untested	High (189)	High (377)	Low (1)	Undefined
Registration with Invalid Email Format	C_2	Untested	High (144)	High (144)	Undefined	Undefined
Sort Products by Price	C_11	Untested	Medium (96)	Low (1)	High (144)	High (144)
Add Product to Cart	C_13	Untested	Medium (60)	Medium (34)	Low (1)	High (144)
Filter Products by Category	C_8	Untested	Medium (50)	Low (5)	Low (1)	High (144)
Login with Incorrect Password	C_4	Untested	Medium (34)	Medium (34)	Undefined	Undefined
Update Profile Information	C_25	Untested	Medium (28)	Medium (55)	Undefined	Low (1)
Track Order Status	C_20	Untested	Medium (28)	Medium (34)	Undefined	Medium (21)
Proceed to Checkout	C_16	Untested	Medium (18)	Medium (34)	Low (1)	Undefined
Apply Discount Code	C_15	Untested	Medium (12)	Medium (34)	Low (1)	Low (1)
View Order History	C_19	Untested	Low (7)	Medium (13)	Undefined	Low (1)
Abandon Cart	C_18	Untested	Low (2)	Low (3)	Low (1)	Low (1)
Password Recovery Attempt for Non-existing Account	C_6	Untested	Undefined	Undefined	Undefined	Undefined

Données masquées d'un système en production.

Exemples d'utilisation

Rapports sur les risques

Risk Burndown Chart ou Graphique de mitigation des risques



Données masquées d'un système en production.

Conclusion
Comparaison et bénéfices

Comparaison des approches de tests basés sur les risques (RBT)

	Approche classique	Approche scientifique (modèle de prédiction bayésien)	Approche avancée (modèle en couches)
Intelligibilité / Configuration	+ facile à comprendre et à configurer	- Très complexe à intégrer, difficile à comprendre	+/- Complexe, mais compréhensible, nécessite une documentation limitée
Granularité / Quantifiabilité	+/- facilement quantifiable, mais très peu nuancé	+ Quantification très granulaire	+ Quantification granulaire et intrinsèquement pondérée
Objectivité	- Très subjectif	+/- Tient compte à la fois des données objectives et subjectives	+ Basé sur les données, évalue les risques sous de nombreux angles
Adaptabilité / Automatisation	- Adaptable, mais une réévaluation constante est nécessaire	+/- Partiellement automatisable et réutilisable	+ Fortement automatisable et adaptable

Bénéfices pour l'assurance qualité

Évaluation des risques

Évaluation **plus objective** des risques
Auto amélioration au fil du temps
Communication unifiée des risques
Évolutivité pour les grands systèmes



Processus de test

Stratégie de test **ciblée**
Meilleurs tests de **régression**



Prévention des risques

Prévention **proactive** des défauts
Meilleure prévention des problèmes **récurrents**



Conclusion

Bénéfices pour l'entreprise

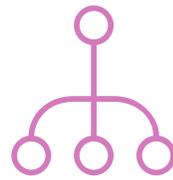
Avantages généraux du RBT

Processus de mise en production plus **stable**
Temps d'**indisponibilité** du système minimisés
Assurance qualité et correction de bugs
rentables
Prévention des atteintes à la **réputation**



Subjectivité réduite

Atténuation globale des risques
Prise de décision plus objective et basée sur les données



Efficacité

Réévaluation implicite des risques à chaque version
Effort d'évaluation des risques **réduit**
Amélioration de l'**efficacité** des ressources



Transparence

Traçabilité des classifications des risques
Évaluation des risques **standardisée**



mgm technology partners France

26, allée Aloyzi Kospicki
38000 Grenoble

Tel +33 4 56 60 00 87

www.mgm-tp.com
info-france@mgm-tp.com

Le modèle de prédiction Bayésien

1. Décision d'exécuter une tâche

Critère de décision :

$$\Lambda(z) \geq_{d_0}^{d_1} \lambda_{BR} \text{ ou simplifié } M(z) \geq_{d_0}^{d_1} \lambda_{BR}$$

Moyenne des avis d'experts :

$$M(z) = \frac{\sum_{e=1}^{|E|} z_e}{|E|}$$

Paramètre du seuil de risque bayésien : $\lambda_{BR} = CR * PR = \frac{C_{10}-C_{00}}{C_{01}-C_{11}} * \frac{P(m_0)}{P(m_1)}$

2. Quantifier la réduction du risque pour la priorisation

Diminution du risque : $\Delta R = R_0 - R_1$

Risque de non-exécution : $R_0 = R(d_0) = C_{00} * P_{00} + C_{01} * P_{01}$

Risque d'exécution : $R_1 = R(d_1) = C_{10} * P_{10} + C_{11} * P_{11}$

Variables et symboles

z_e – Avis d'experts sur le nombre de défauts d'une tâche

Decisions:

d_0 – omettre une tâche

d_1 – exécuter une tâche

Résultats :

m_0 – Aucun défaut détecté

m_1 – Au moins un défaut détecté

Facteurs de coût :

$C_{00} = C(d_0, m_0)$ – Aucun coût lorsqu'on omet la tâche et qu'aucun défaut n'est détecté

$C_{01} = C(d_0, m_1)$ – Coût le plus élevé dû à une détection manquée d'un défaut, nécessitant des corrections coûteuses après la livraison et mettant en péril la réputation

$C_{10} = C(d_1, m_0)$ – Coût pour une exécution inutile de la tâche en l'absence de défaut

$C_{11} = C(d_1, m_1)$ – Coût pour l'exécution de la tâche et la réparation du défaut

Probabilités:

$P(m_0)$ – aucun défaut détecté précédemment

$P(m_1)$ – au moins un défaut détecté précédemment

$P_{00} = P(d_0, m_0)$ – Probabilité d'omettre correctement une tâche

$P_{01} = P(d_0, m_1)$ – Probabilité d'omettre incorrectement une tâche

$P_{10} = P(d_1, m_0)$ – Probabilité d'exécuter incorrectement une tâche

$P_{11} = P(d_1, m_1)$ – Probabilité d'exécuter correctement une tâche

Le modèle de prédiction Bayésien- Exemple

1. Décision d'exécuter une tâche

Critère de décision :

$$\Lambda(z) \geq_{d_0}^{d_1} \lambda_{BR} \text{ or simplified } M(z) \geq_{d_0}^{d_1} \lambda_{BR}$$

Moyenne des avis d'experts :

$$M(z) = \frac{\sum_{e=1}^{|E|} z_e}{|E|} = \frac{0.5+0.6+0.7}{3} = 0.6$$

Paramètre du seuil de risque bayésien : $\lambda_{BR} = CR * PR = \frac{C_{10}-C_{00}}{C_{01}-C_{11}} * \frac{P(m_0)}{P(m_1)} = \frac{2-0}{20-5} * \frac{0.6}{0.4} = \frac{2}{15} * 1.5 = 0.2$

2. Quantifier la réduction du risque pour la priorisation

Diminution du risque :

$$\Delta R = R_0 - R_1 = 4 - 1.6 = 2.4$$

Risque de non-exécution : $R_0 = R(d_0) = C_{00} * P_{00} + C_{01} * P_{01} = 0 * 0.3 + 20 * 0.2 = 0 + 4 = 4$

Risque d'exécution :

$$R_1 = R(d_1) = C_{10} * P_{10} + C_{11} * P_{11} = 2 * 0.3 + 5 * 0.2 = 0.6 + 1 = 1.6$$

Variables et symboles

$z_e \in \{0.5, 0.6, 0.7\}$

Décisions:

d_0 - omettre une tâche

d_1 - exécuter une tâche

Résultats :

m_0 - Aucun défaut détecté

m_1 - Au moins un défaut détecté

Facteurs de coût :

$C_{00} = C(d_0, m_0) = 0$

$C_{01} = C(d_0, m_1) = 20$

$C_{10} = C(d_1, m_0) = 2$

$C_{11} = C(d_1, m_1) = 5$

Probabilités::

$P(m_0) = 0.6$

$P(m_1) = 0.4$

$P_{00} = P(d_0, m_0) = 0.3$

$P_{10} = P(d_1, m_0) = 0.3$

$P_{01} = P(d_0, m_1) = 0.2$

$P_{11} = P(d_1, m_1) = 0.2$