Fiabilité du logiciel

... Oui, quelquefois

le système plante …

Motivations

- Le logiciel est un composant essentiel de plusieurs systèmes de sécurité critique.
- Ces systèmes dépendent de la fiabilité des composants logiciels.
- Comment pouvons-nous décider si le logiciel est prêt à être distribué après le test de système ?
- Comment pouvons-nous décider si un composant acquis ou développé est acceptable ?

Qu'est-ce que la fiabilité ?

- **IEEE-Std-729-1991** :

  "La fiabilité logicielle est définie comme la probabilité de ne pas avoir une défaillance de service pour une *période de temps* spécifique dans un *environnement* spécifique."

- **ISO9126:**

  "La fiabilité est la capacité du produit logiciel à maintenir un niveau de performance spécifique lorsqu'il est utilisé sous des conditions spécifiques."

La fiabilité en général

- → Probabilité de manque de défaillance de service pour un temps spécifique dans un environnement spécifique pour une intention donnée.
- La signification est différente selon le système et les utilisations de ce système.
- Informellement, la fiabilité est une mesure qui indique à quel point le logiciel fournie les services attendus par les usagers.
- Quantification : Nombre, sévérité des défaillances.

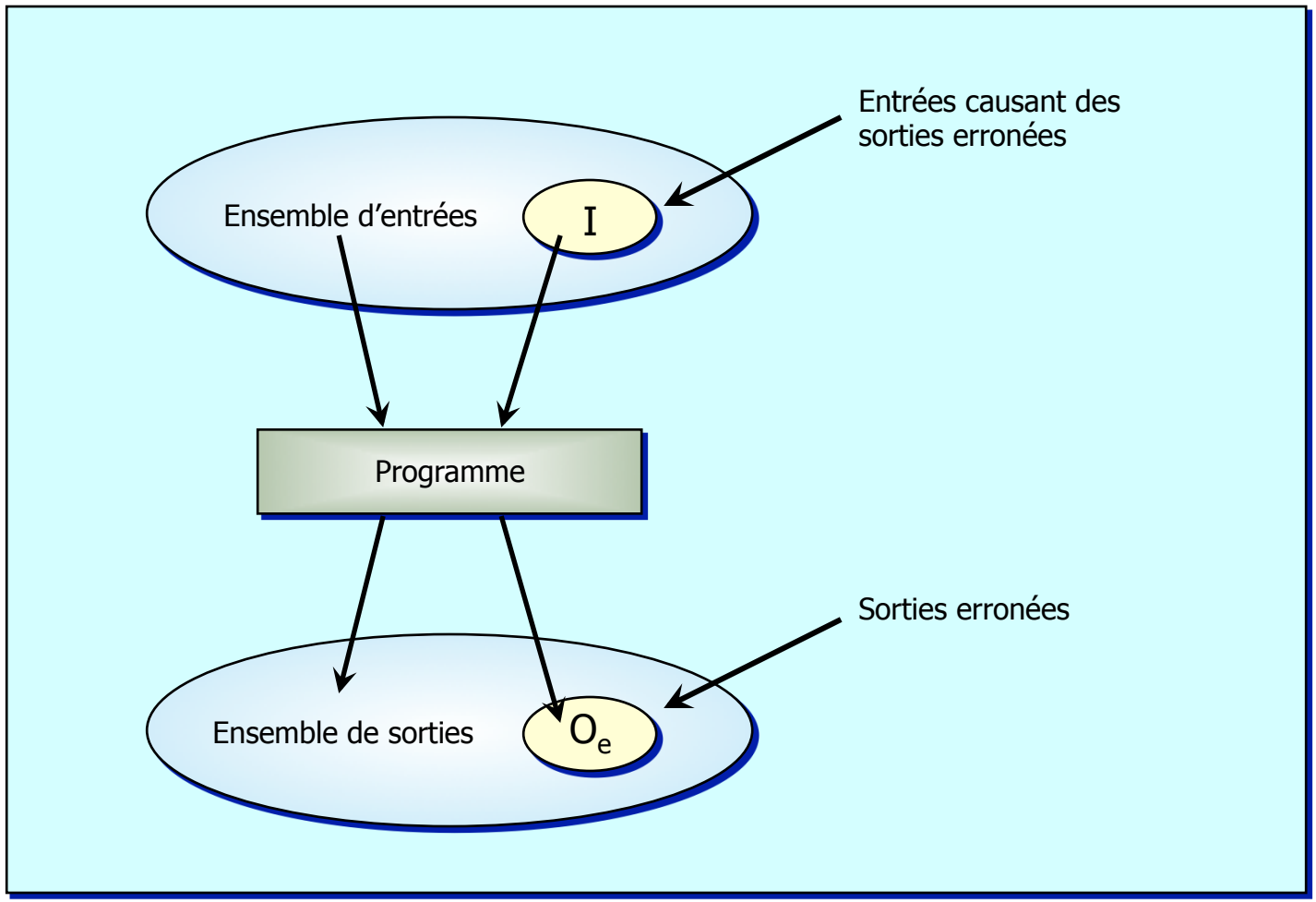
La fiabilité du logiciel

- Ne peut être définie de manière absolue.
  - Une mesure de fiabilité citée hors contexte n'est pas significative.
- Requiert un profil opérationnel pour sa définition.
  - Le profil opérationnel définit le modèle escompté pour l'usage du logiciel.
- Doit considérer les conséquences fautives.
  - Les erreurs ne sont pas toutes également graves. Le système sera perçu d'autant moins fiable si les erreurs y sont plus graves.

Faute et défaillance

- Une défaillance correspond à l'observation d'un comportement imprévu au moment d'une exécution (par un usager).
- Une faute est une caractéristique statique de logiciel qui suscite la panne.
- Les fautes ne suscitent pas nécessairement des défaillances.
- Si un usager ne remarque pas une défaillance, est-ce alors une défaillance ?
  - Souvenez-vous que la plupart des usagers ne connaissent pas la spécification du logiciel.

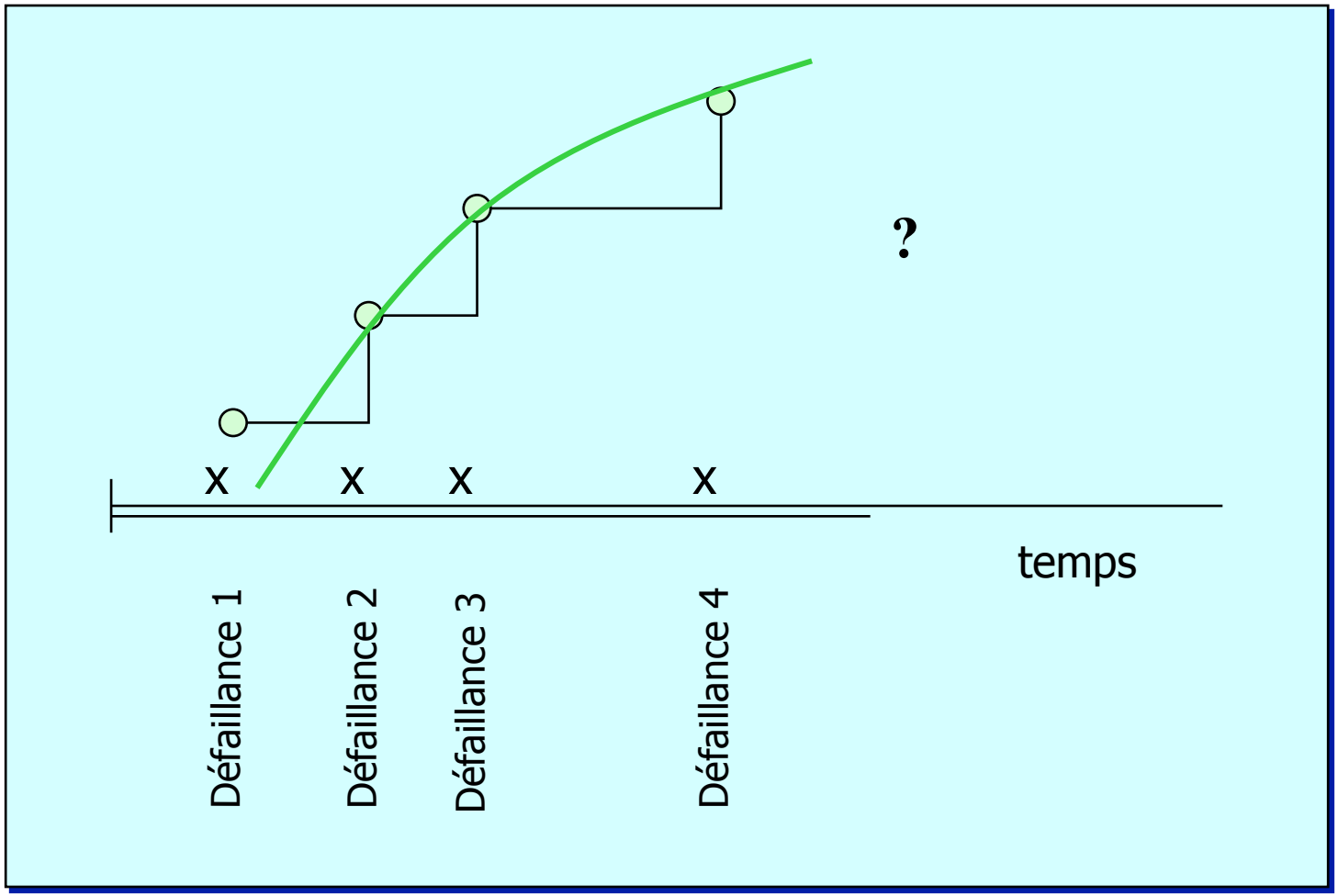
Topologie d'entrée/sortie



Comment pouvons-nous mesurer et modéliser la fiabilité ?

- Le nombre cumulatif de défaillances pour un temps t.
- *L'intensité de la défaillance* : Le nombre de défaillances par unité de temps.
- *Intervalle moyen avant une défaillance (*Mean-Time-To-Failure (MTTF)) : La valeur moyenne de l'intervalle de temps de l'inter-défaillance.
- On ne peut être certain du moment où le logiciel aura sa prochaine défaillance => le comportement de défaillance est modélisé comme processus aléatoire.

Modéliser le # cumulatif de défaillances



Plus utile pour un usager
- Probabilité d'échec pour une période donnée (mission)
- MTTF

Croissance de la fiabilité

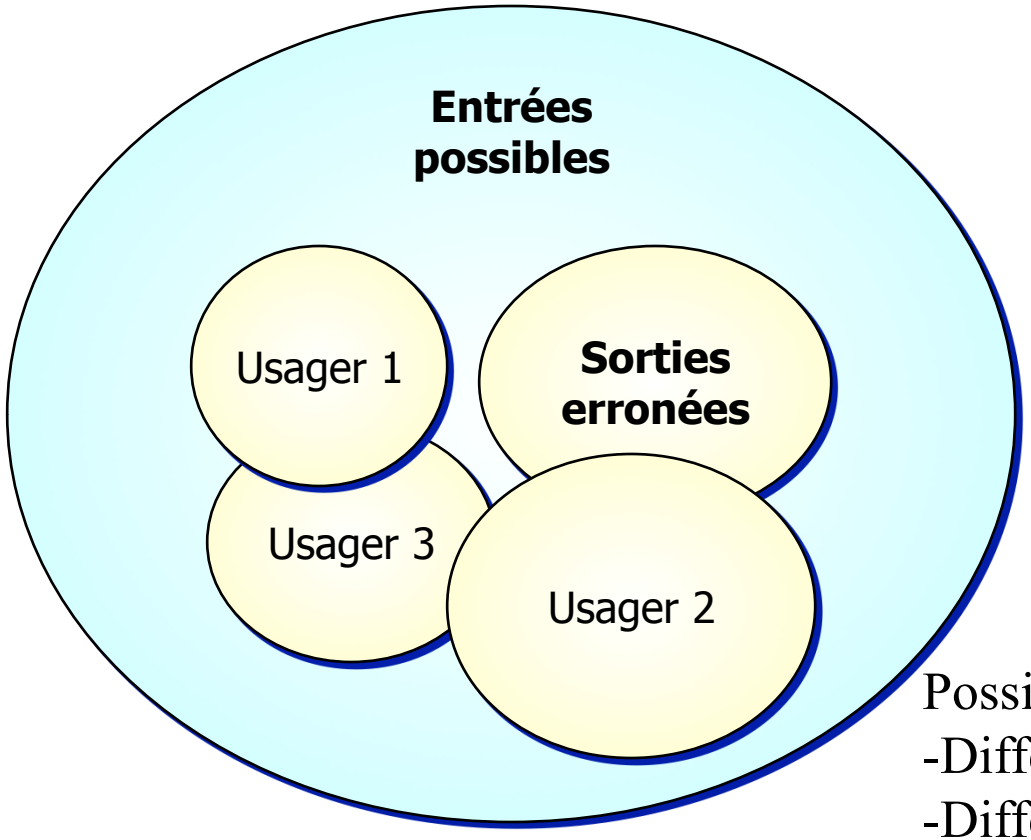
- La fiabilité s'améliore quand les fautes du logiciel qui se produisent dans les parties les plus fréquemment utilisées de celui-ci sont supprimées.
- Supprimer x% des fautes du logiciel ne conduira pas nécessairement à x% d'amélioration de la fiabilité.
- Une étude a montré que, supprimer 60% des défaillances du logiciel mène à une amélioration de la fiabilité de 3%.
- Supprimer les fautes qui ont des conséquences sérieuses est l'objectif le plus important.

Problèmes avec la quantification ?

- La norme américaine pour l'équipement sur des avions civils :
  - [Une défaillance critique doit être] si improbable qu'[elle] doit être considérée ne pouvoir jamais se produire, à moins qu'un jugement d'ingénierie ne requiert [sa] considération.
- *Requis typique pour systèmes de vol critiques* :

  10^{-9} : probabilité de défaillance par heure de vol basée sur un vol d'une durée moyenne de 10 heures.
- *Problème 1* : un tel niveau de fiabilité ne peut pas être démontré – requerrait trop de temps de test.
- *Problème 2* : La fiabilité est relative à l'usage du système or cet usage peut être compliqué à déterminer (voir prochain slide)

Modèles d'utilisation



Possiblement
-Différentes utilisations
-Différents sites
-Différentes catégories d'utilisateurs

Fiabilité et qualité

- La fiabilité est normalement plus importante que d'autres aspects de qualité.
- Un logiciel non fiable n'est pas utilisé.
- Difficile d'améliorer la qualité des systèmes non fiables.
- La défaillance d'un logiciel coûte souvent beaucoup plus cher que le coût du système.
- Les coûts de perte de données sont très élevés.

Métriques de fiabilité

- Probabilité de défaillance sur demande (POFOD: Probability Of Failure On Demand)
  - C'est une mesure de la probabilité que le système ait une défaillance quand une requête de service sera faite.
  - POFOD = 0.001 signifie 1 sur 1000 requêtes de service résulte en une défaillance.
  - Applicable pour une sécurité critique ou des systèmes continus.
- Taux de production de défaillances (ROCOF: Rate Of Failure Occurrence)
  - Fréquence de production de comportement inattendu.
  - Un ROCOF de 0.02 signifie qu'il y aura 2 défaillances probables dans chaque 100 unités de temps opérationnel.
  - Applicable pour des systèmes d'exploitation, des systèmes de traitement de transactions.

Métriques de fiabilité

- Temps moyen entre deux défaillances (MTTF: Mean Time To Failure)
  - Mesure du temps entre les défaillances observées.
  - MTTF de 500 signifie que le temps entre les défaillances est de 500 unités de temps.
  - Applicable pour systèmes avec longues transactions e.g. systèmes CAD (Computer Aided Design).
- Disponibilité (AVAIL: AVAILability)
  - Mesure de la disponibilité d'utilisation du système. Prends en considération les réparations et les temps de réinitialisation.
  - Disponibilité de 0.998 signifie que le logiciel est disponible pour 998 sur 1000 unités de temps.
  - Applicable pour des systèmes continuellement en marche e.g. systèmes de communication téléphonique.

Mesure de la fiabilité

- Mesure du nombre de défaillances du système pour un certain nombre d'entrées de données.
  - Utilisée pour estimer POFOD.
- Mesure du temps (ou nombre de transactions) entre les défaillances du système.
  - Utilisée pour estimer ROCOF et MTTF.
- Mesure du temps de mise à zéro après la défaillance.
  - Utilisée pour estimer AVAIL.

Unité de temps

- Les unités de temps dans la mesure de la fiabilité doivent être sélectionnées soigneusement. Elles ne sont pas les mêmes pour tous les systèmes.
- Le temps d'exécution brut (pour les systèmes de communication téléphonique).
- Le temps calendaire (pour les systèmes continues, par exemple les systèmes d'alarme).
- Le nombre de transactions (pour les systèmes qui sont utilisés sur demande, ex. ATM).

Conséquences de défaillances

- Les mesures de fiabilité NE tiennent PAS compte des conséquences des défaillances.
- Les fautes transitoires peuvent ne pas avoir de conséquences réelles mais d'autres fautes peuvent causer des pertes de données ou la corruption et la perte de service de système.
- Les conséquences de défaillances peuvent être nécessaires pour identifier différentes classes de défaillances et utiliser différentes mesures pour chacune d'elles.

Une classification

Classe de défaillance	Description
Transitoire	se produit seulement avec certaines entrées (sous des circonstances passagères et transitoires)
Permanente	se produit avec toutes les entrées.
Récupérable	le système peut récupérer sans l'intervention de l'opérateur.
Non-récupérable	l'intervention de l'opérateur est nécessaire pour récupérer.
Non-corrompue	la défaillance n'a pas corrompu l'état du système ou des données.
Corrompue	la défaillance a corrompu l'état du système et des données.

Spécification

- Les besoins de la fiabilité sont rarement seulement exprimés de manière quantitative et vérifiable.
- Pour vérifier les métriques de fiabilité, un profil opérationnel doit être spécifié comme faisant partie du plan de test.
- La fiabilité est dynamique – les spécifications de la fiabilité reliées au code source sont insignifiantes.
  - Pas plus que N fautes/1000 lignes (A-t-on vraiment détecté toutes les fautes?).
  - Utile seulement pour une analyse du processus de post-distribution.

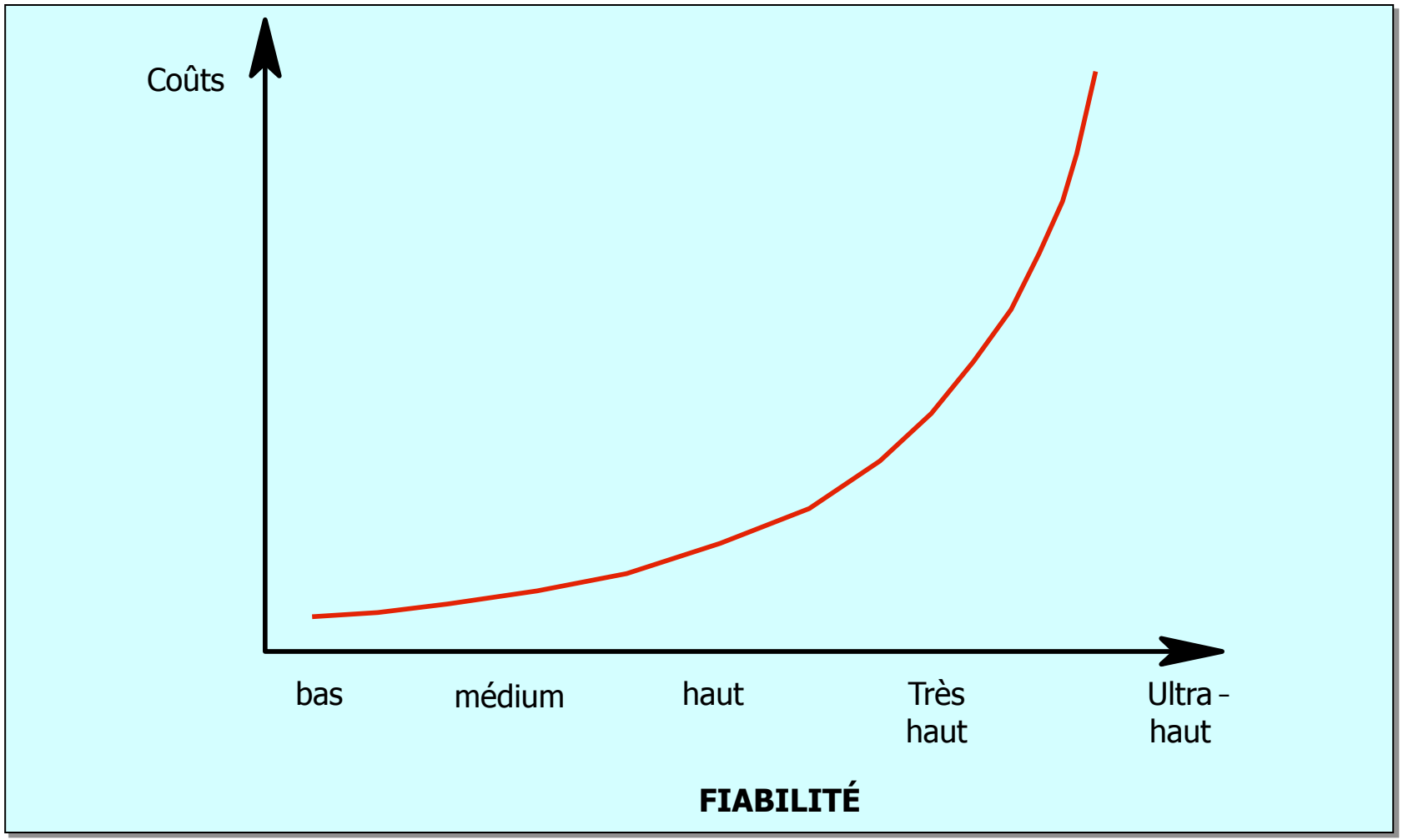
Validation des spécifications

- Il est impossible ***de valider empiriquement*** des spécifications de très haute fiabilité.
- Prenons l'exemple d'un réseau ATM devant durer 8 ans qui avec 1000 bornes à environ 300 transactions par jour effectue en moyenne 300 000 transactions par jour et approximativement 100 000 000 transactions par année. Supposons que pour s'assurer qu'on n'aura jamais de corruption de la base de données, on ait spécifié un POFOD (en rapport avec la BDD) d'au plus 1/200 000 000 000
  - Si avec une simulation du réseau, une transaction prend 1 seconde, alors simuler des transactions d'une journée prend 300 000s = 3.5 jours.
  - Une validation empirique d'une telle valeur de POFOD prendrait énormément plus de temps que la durée de vie du système.

Compromis

- Compte tenu des coûts très élevés de la fiabilité absolue (faire ce qu'il faut pour, s'assurer que), il peut parfois être plus efficace d'accepter jusqu'à un certain point la non fiabilité (et de payer pour les coûts de défaillances).
- Facteurs de décision : social, d'affaire et politique :
  - Une réputation de produits non fiables peut faire perdre de futures affaires
- Dépendamment de la « criticalité » des produits en question): en général, s'ils concernent les affaires de particuliers, une fiabilité modeste peut être adéquate.

Coûts



Test statistique

- Tester le logiciel pour sa fiabilité plutôt que pour la détection de fautes.
- La sélection des données de test devrait suivre le profil d'utilisation prédit pour le logiciel.
- Mesurer le nombre d'erreurs permet de prédire la fiabilité du logiciel.
- Un niveau acceptable de fiabilité devrait être spécifié et le logiciel devrait être testé et corrigé jusqu'à ce que ce niveau de fiabilité soit atteint.

Tester un logiciel

- Déterminer le *profil opérationnel* du logiciel.
- Générer un ensemble de données de test correspondant à ce profil.
- Appliquer les tests, mesurer le total de temps d'exécution entre chaque défaillance.
- Après qu'un nombre statistique valide de tests ait été exécuté, la fiabilité peut être mesurée.

Difficultés

- Incertitude dans le profil opérationnel.
  - C'est un problème surtout pour les nouveaux systèmes sans histoire opérationnelle. C'est moins le cas pour les systèmes de remplacement.
- Coûts élevés de génération de profil opérationnel.
  - Les coûts sont très dépendants du traitement de l'information dans l'organisation à laquelle correspond le profil.
- Incertitude statistique quand un haut niveau de fiabilité est spécifié.
  - Difficulté pour estimer le niveau de confiance dans le profil opérationnel.
  - Le modèle de logiciel utilisé peut changer avec le temps.

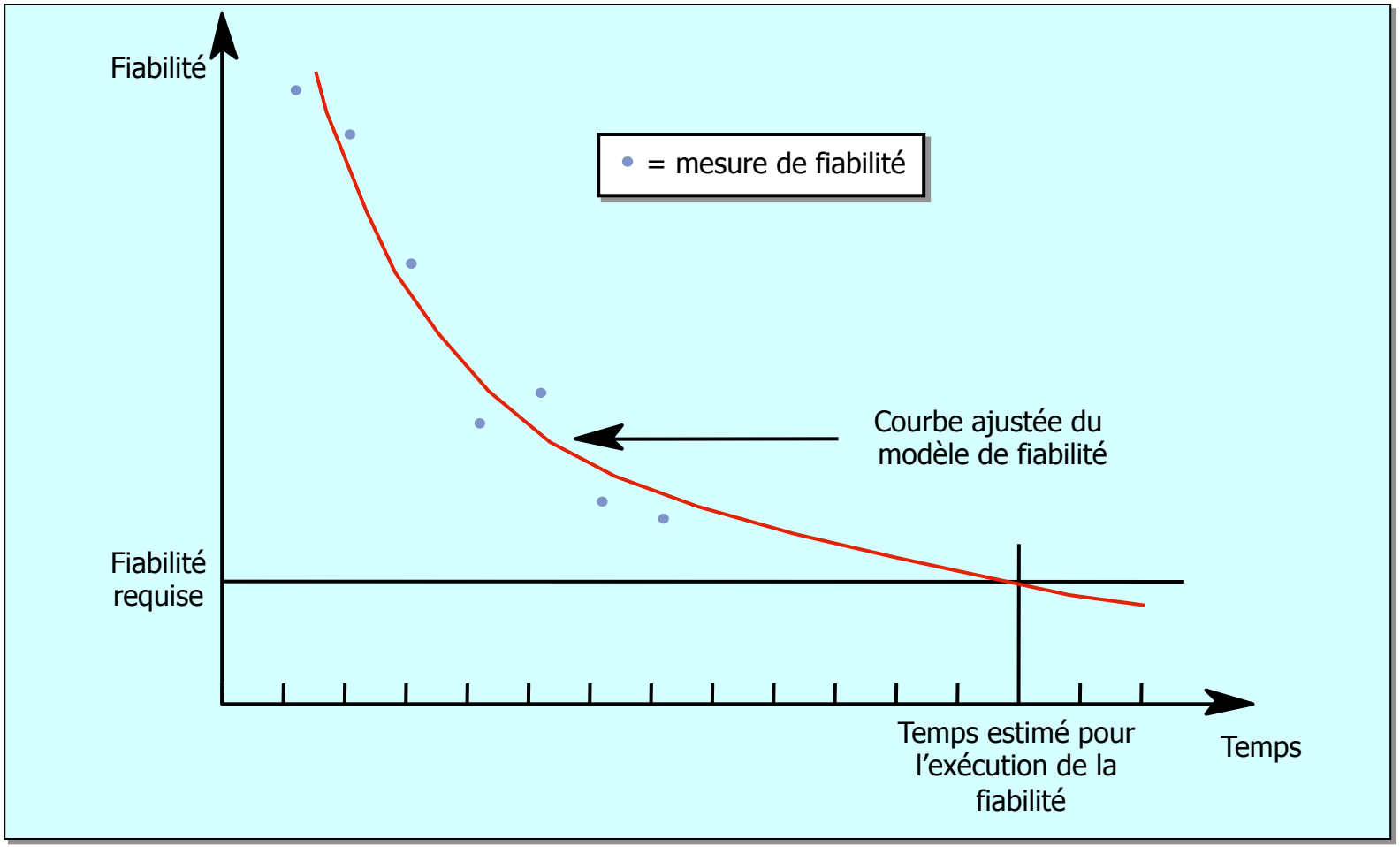
Modèle croissant de la fiabilité

- Le modèle croissant est un modèle mathématique de la fiabilité du système qui change quand il est testé et que les fautes sont supprimées.
- Utilisé comme une moyen de prédiction de fiabilité en extrapolant des données courantes.
- Dépend de l'utilité de tester statistiquement la mesure de fiabilité d'une version du système.

Modèles de fiabilité

- Plusieurs modèles de croissance de fiabilité ont été proposés.
- Pas de modèle de croissance universellement applicable.
- La fiabilité devrait être mesurée et les données observées devraient être adaptées à plusieurs modèles.
- Le modèle le mieux adapté devrait être utilisé pour la prédiction de la fiabilité.

Modèles et prédiction



Fiabilité matérielle

- La théorie de fiabilité est plus développée et établie.
- Un composant matériel peut avoir une défaillance due à une erreur de conception, à une fabrication de pauvre qualité, à une surcharge momentanée, à une détérioration, etc.
- La nature aléatoire des défaillances permet l'estimation de la fiabilité en utilisant des modèles probabilistes.
- Peut-on l'adapter au logiciel ? Pas de détérioration physique, la plupart des programmes fournissent toujours la même réponse à la même donnée, la "fabrication" de code est insignifiante, etc.

L'ingénierie de la fiabilité du logiciel

- Aucune méthode de développement ne peut garantir un logiciel totalement fiable => champ important en pratique!
- Un ensemble de techniques de modèles statistiques.
- Permet à la fiabilité atteinte d'être *estimée* ou *prédite*, quantitativement et objectivement.
- Basée sur l'observation des défaillances de système durant le test de système et l'usage opérationnel.
- Utilise les théories de fiabilité générale, mais n'y est pas réduite.

Applications

- Quelle fiabilité courante pour le programme/composant?
- Basées sur la fiabilité courante d'un composant acheté/réutilisé : pouvons-nous l'accepter ou devrions-nous le rejeter ?
- Basées sur la fiabilité courante du système : pouvons-nous arrêter de tester et commencer à distribuer ?
- Quand est-ce que le système sera fiable, si nous continuons à le tester encore quelque temps ?
- Quand est-ce que l'objectif de fiabilité sera atteint ?
- Combien de défaillances se produiront après déploiement (et quand)? Sert dans la planification des ressources à allouer pour le support des clients

Détour … Mathématique

Variable aléatoire

- Une variable aléatoire X est une fonction définie dans un espace d'échantillonnage S qui associe un nombre réel x à chaque échantillon possible de S : $X(e) = x$.
- Si S est fini ou dénombrable, on dit que X est discrète.
- Si X peut être n'importe quelle valeur dans un intervalle donné, nous disons que X est une variable aléatoire continue.

Variable aléatoire discrète

- Si S est dénombrable, X(e) est dénombrable.
- Probabilité d'événement : $P\{e: X(e) = x, e \in S\}$
- Fonction de masse (Probability mass function (pdf)) :

$$p(x_i) \geq 0 \quad \sum_{x_i} p(x_i) = 1$$

- Fonction de répartition (Cumulative distribution function (cdf)) :

$$P\{X \leq x\} = \sum_{x_i \leq x} p(x_i)$$

Espérance et variance

$$E[X] = \sum_{x} x p(x)$$

$$Var(X) = E\left[\left(X - E(X)\right)^2\right]$$

Variables aléatoires continues

La pdf de la variable aléatoire continue X est une fonction :

$$f(x)$$

Avec :

$$\forall x: f(x) \geq 0 \quad \int_{-\infty}^{+\infty} f(x)dx = 1$$

Fonction de répartition:

$$F(\mathrm{x}) = \int_{-\infty}^{x} f(y)dy$$

Espérance et variance

$$E[X] = \int_{-\infty}^{+\infty} x f(x) dx$$

$$Var(X) = E\left[(X - E[X])^2\right]$$

Probabilités conditionnelles

Nous sommes intéressés en un sous-ensemble B de l'ensemble de l'univers de S, en d'autres mots, l'échantillon spatial est essentiellement B. Disons que $P(B)>0$ et disons que A est un sous-ensemble de S, nous définissons la probabilité de A relative au nouvel échantillon spatial B :

$$P(A|B) = P(A \cap B) / P(B)$$

Distributions conditionnelles

Discrète :

$$p(x_i|x_j) = \mathrm{P}(\mathrm{X} = x_i|\mathrm{X} = x_j) = \frac{\mathrm{P}(\mathrm{X} = x_i\ \mathrm{X} = x_j)}{\mathrm{P}(\mathrm{X} = x_j)}$$

Continue :

$$f(x|y) = \frac{f(x,y)}{f(y)}$$

Avec :

$$f(\mathrm{x}) = \int_{-\infty}^{+\infty} f(x,y)\,dy$$

Processus stochastique

Un processus stochastique est une collection de variables aléatoires :

$$X_t \text{ ou } X(t)$$

où t est un ensemble de l'index approprié. En d'autres mots, t peut être une unité de temps discrète. L'ensemble de l'index est $T=[0,1,2,3,...]$ ou il peut être un point dans un intervalle de temps continu, alors l'ensemble de l'index est égal à :

Fiabilité

La variable aléatoire d'intérêt est le temps de défaillance T.
Nous concentrons notre intérêt sur la probabilité que le temps entre les défaillances soit de quelques unités de temps.
Le temps de défaillance T étant donné un intervalle Δt

$$T \in (t\,,\, t + \Delta t)$$

En d'autres mots :

$$P(t \leq T \leq t + \Delta t) = f(t)\Delta t \;=\; F(t + \Delta t) - F(t)$$

La pdf f est aussi appelée fonction de densité de la défaillance.

La fonction de fiabilité (Reliability Function R(t))

Puisque que T est seulement définie pour des valeurs positives :

$$F(t) = P(0 \leq T \leq t) = \int_0^t f(y)dy$$

est la probabilité d'observer une défaillance avant t. La probabilité de succès au temps t, R(t), i.e. le temps de défaillance est plus grand que t :

$$R(t) = P(T > t) = 1 - F(t) = \int_t^{+\infty} f(y)dy$$

Taux de défaillance

Le taux de défaillance est la probabilité qu'il se produise une défaillance par unité de temps dans un intervalle Δt, sachant qu'il n'y en a pas eu avant t :

$$\frac{P(t \le T < t + \Delta t | T > t)}{\Delta t} = \frac{P(t \le T < t + \Delta t)}{\Delta t\, P(T > t)} = \frac{F(t + \Delta t) - F(t)}{\Delta t\, R(t)}$$

Taux aléatoire

Le taux aléatoire est la limite quand l'intervalle de taux de défaillance approche zéro :

$$z(t) = \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t \, R(t)} = \frac{f(t)}{R(t)}$$

Étant donné que le système survit jusqu'à t, $z(t)$, est le taux de défaillance instantané, i.e. $z(t)dt$ est la probabilité qu'un système d'âge t fera défaut dans un petit intervalle $[t, t+dt[$.

R(t) et *z(t)*

Par définition : $$z(t) = \frac{dF(t)}{dt} * \frac{1}{R(t)}$$

Puisque : $R(t) = 1 - F(t)$, ce qui signifie $$\frac{dR(t)}{dt} = - \frac{dF(t)}{dt}$$

Sachant que $\frac{d}{dx}\ln(f(x)) = \frac{f'(x)}{f(x)}$

On peut dériver $$\lg R(t) = - \int_0^t z(x)dx + c$$

Puisque le système est normalement bon à $t=0$, $R(0)=1$ ➔ c = 0

$$R(t) = \exp\left[- \int_0^t z(x)dx \right]$$

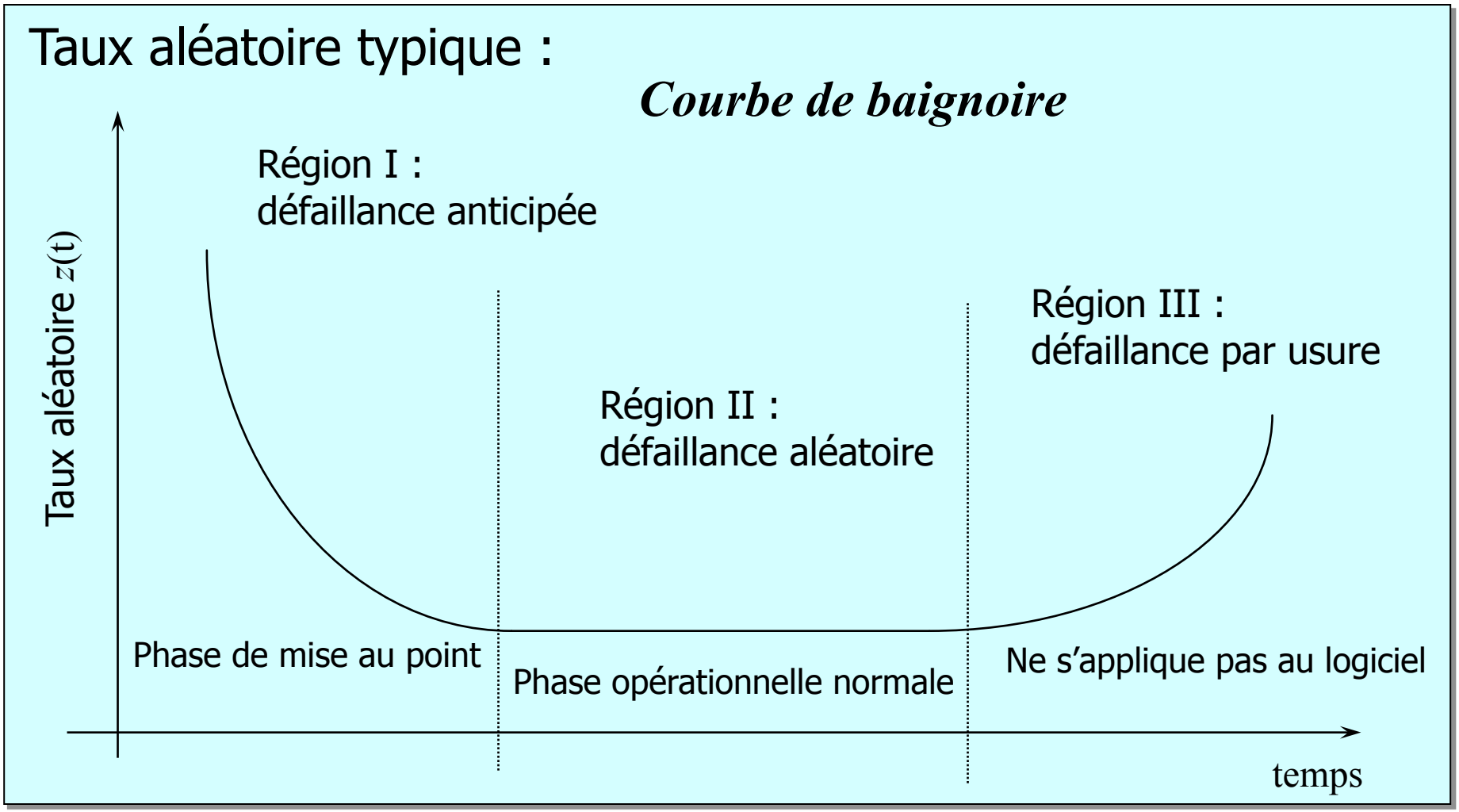
R(t), z(t), f(t)

$$R(t) = \exp\left[- \int_0^t z(x)dx \right]$$

$$f(t) = z(t) \exp\left[- \int_0^t z(x)dx \right]$$

Temps supplémentaire de taux aléatoire

Taux aléatoire typique :



Exemple

Taux aléatoire constant

$$z(t) = \lambda$$

$$f(t) = \lambda \ e^{-\lambda t}$$

$$R(t) = e^{-\lambda \ t}$$

Taux aléatoire linéaire croissant

$$z(t) = K \lambda$$

$$f(t) = Kt \, e^{-K t^2/2}$$

$$R(t) = e^{-K \ t^2/2}$$

Vie estimée

On se reporte souvent à l'intervalle moyen avant la défaillance (Mean Time To Failure (MTTF)) :

$$\mathrm{MTTF} = \mathrm{E}[\mathrm{T}] = \int_0^\infty t f(t)\,dt$$

En terme de fiabilité :

$$\mathrm{MTTF} = \int_0^\infty \mathrm{R}(t)\,dt$$

Exemples de MTTF

Taux aléatoire constant :

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda}$$

Taux aléatoire linéaire croissant :

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty e^{-Kt^2/2} dt = \frac{\Gamma\left(\frac{1}{2}\right)}{2\sqrt{K/2}} = \sqrt{\frac{\pi}{2K}}$$

Un autre point de vue

Au lieu de la variable aléatoire T, le temps avant la prochaine défaillance, nous pouvons modéliser en utilisant le nombre de défaillances qu'un système éprouve avant un temps t.

De manière évidente, le nombre de défaillances éprouvées avant un temps t est une variable aléatoire; c'est en fait un ***processus aléatoire***, puisqu'une fois le temps t fixé, le nombre de défaillances éprouvées est une variable aléatoire.

Intensité de la défaillance

Disons que **M(t)** est un processus aléatoire représentant le nombre cumulatif de défaillances au temps t, cela signifie que la valeur moyenne est :

$$\mu(t) = E\left[M(t)\right]$$

La fonction $\lambda(t)$, l'intensité de la défaillance, est :

$$\lambda(t) = \frac{d\mu(t)}{dt} = \frac{d}{dt}\left(E\left[M(t)\right]\right)$$

Propriétés de l'intensité de la défaillance

L'intensité de la défaillance représente le nombre anticipé de défaillances par unité de temps.

Le nombre de défaillances qui se produiront entre t et $t + \Delta t$ peut être estimé par $\lambda\,(t)\,\Delta t$

L'intensité de la défaillance, devrait diminuer afin d'obtenir une croissance dans $R(t)$.

Classification des modèles de fiabilité

- **Domaine temporel** :

  horloge murale versus temps d'exécution.

- **Catégorie** :

  le nombre total de défaillances qui peut être produit dans un temps infini (fini ou infini).

- **Type** :

  la distribution du nombre de défaillances produit au temps t (Poisson ou binominal).

Type Poisson

La distribution des défaillances produite au temps t est un processus de Poisson. En d'autres mots, si :

$$t_0 = 0,\ t_1,\ t_2,\ldots,t_n = t$$

est une partition de l'intervalle $[0,t]$, nous avons un Processus de Poisson si f_i avec $i = 1,2,\ldots,n$, le nombre de fautes dans le i^{ème} intervalle sont des variables indépendantes de Poisson avec

$$E[f_i] = \mu(t_i) - \mu(t_{i-1})$$

Classification

Si $\mu(t)$ est une fonction linéaire de temps, nous disons que $M(t)$ est un processus homogène de Poisson (Homogenous Poisson Process (HPP)).

Si $\mu(t)$ est une fonction non linéaire, nous disons que $M(t)$ est un processus non-homogène de Poisson.

Processus binominal

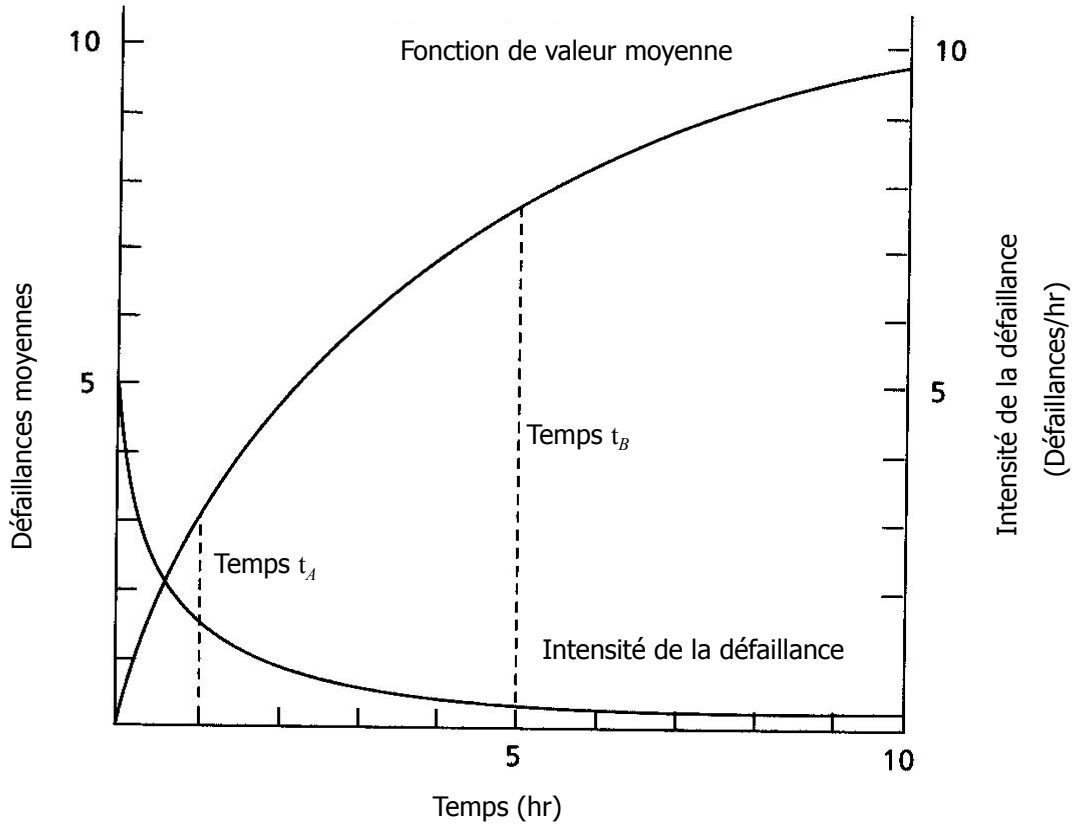
- Il y a un nombre fixe de fautes (N) dans le logiciel au commencement du temps pendant lequel le logiciel est observé.
- Quand une faute est détectée, elle est immédiatement supprimée.
- Si Ta est la variable aléatoire indiquant le temps de la défaillance de la faute a, alors les Ta pour $a=1,2,3, \ldots n$ sont indépendamment des variables aléatoires identiquement distribuées (tout comme les fautes restantes!).

Fin du detour …

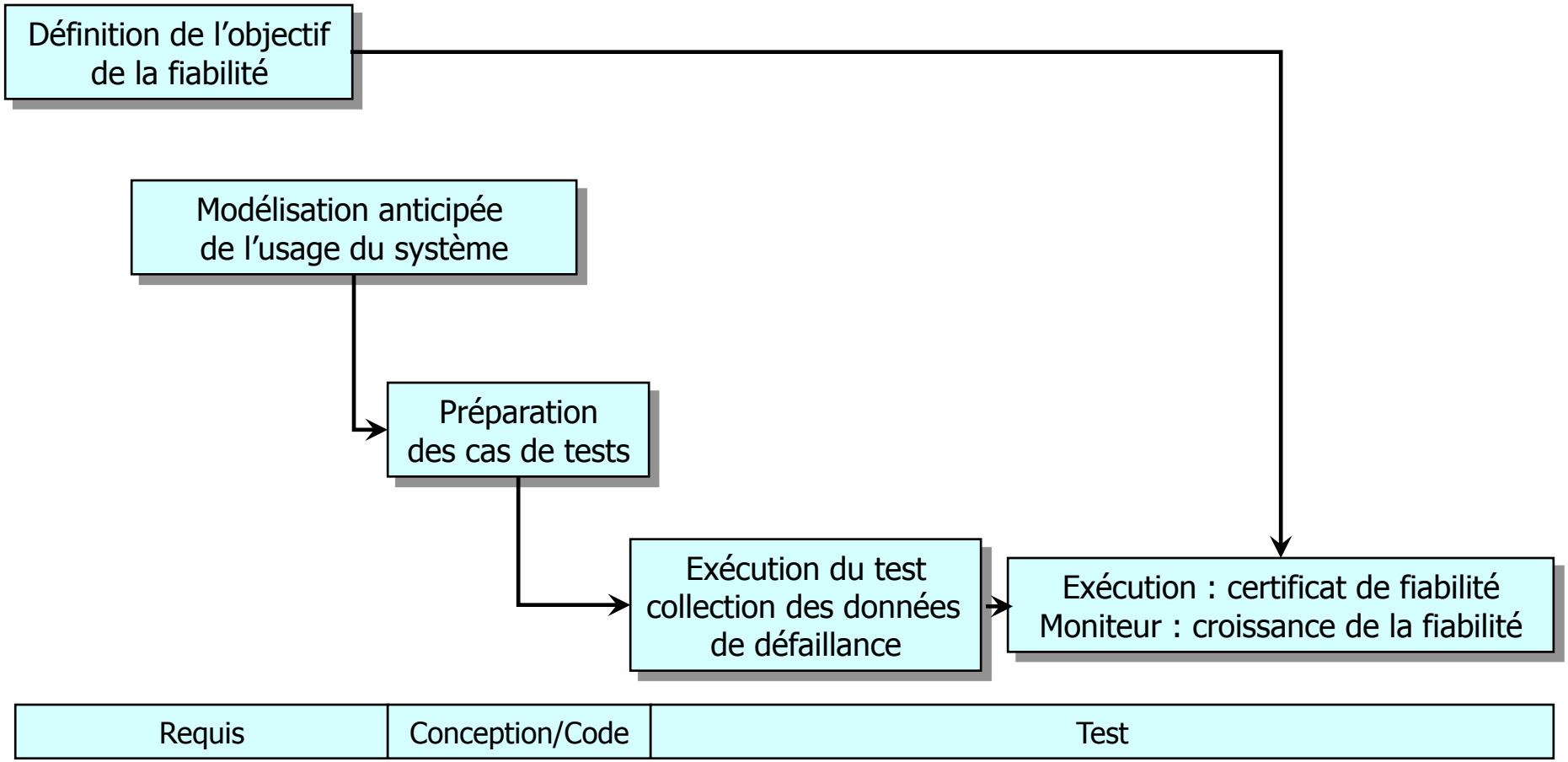
Intensité de la défaillance

- $\mu(t) = E[M(t)]$, avec $M(t)$ le processus aléatoire indiquant le nombre cumulatif de défaillances au temps t, nous indiquons $\mu(t)$ comme sa fonction de valeur moyenne (sur plusieurs instanciations indépendantes du logiciel).
- $\lambda(t)$: L'intensité de la défaillance, le nombre moyen de défaillances à l'unité de temps au temps t, taux instantané de changement du *nombre anticipé de défaillances* ($\mu(t)$).
- La croissance de la fiabilité suggère : $d\lambda(t)/dt < 0$.

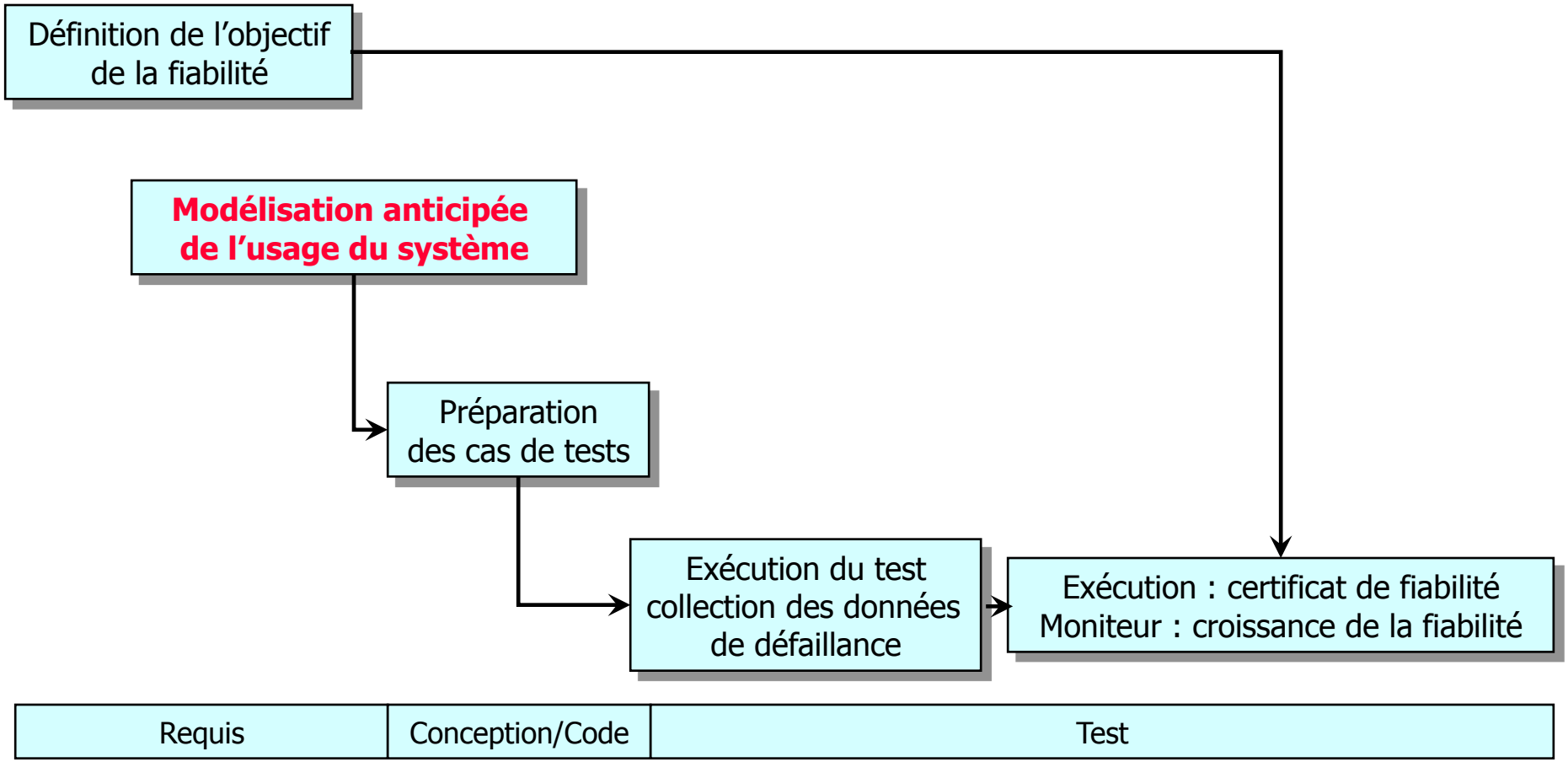
Représentation graphique



Processus d'ingénierie de la fiabilité du logiciel



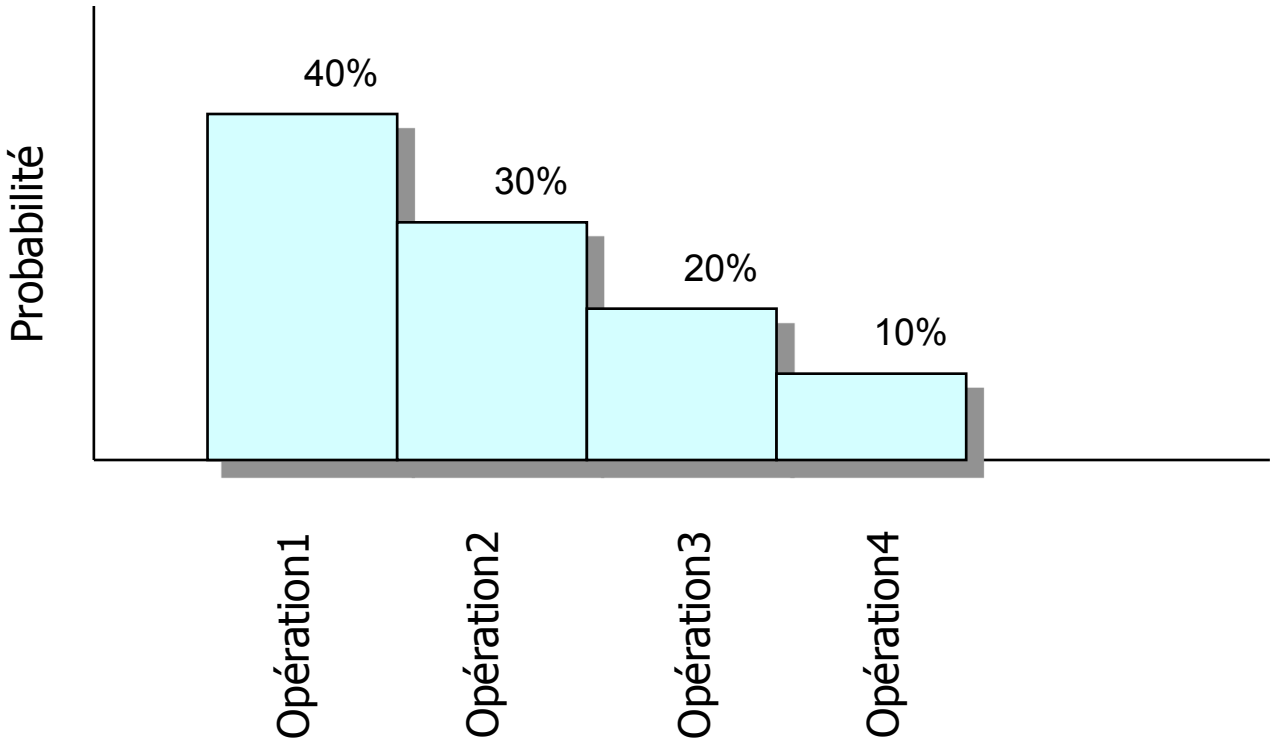
Processus d'ingénierie de la fiabilité du logiciel



Modèle anticipé de l'usage du système

- La définition de la fiabilité suppose un *environnement spécifié.*
  - ⇨ Pour anticiper la fiabilité « sur site » durant le test du système, nous devons tester dans des conditions "similaires aux conditions sur site".
- *Modéliser comment les usagers emploieront le logiciel* : environnement, type d'installation, distribution des données dans l'espace des données.
- En accord avec le modèle utilisé, les cas de test sont choisis *aléatoirement.*
- Un exemple de modèle utilisé
  - *Profil opérationnel* (Musa) : Ensemble des opérations du système et leurs probabilités de se produire.

Profil opérationnel



Opérations

- Tâche logique majeure du système, de courte durée, qui retourne le contrôle au système lorsqu'elle est complétée et pour laquelle le processus est substantiellement différent d'autres opérations.
  - Majeure : reliée aux besoins fonctionnels (similaire aux cas utilisés).
  - Logique : pas limitée au logiciel, au matériel, aux usagers, aux machines.
  - Court : 100s-1000s opérations par heure sous des conditions de charge normale.
  - Différente : susceptible de contenir des fautes différentes.
- Dans un système OO, une opération ~ 'use case'.

Exemples

- Commande exécutée par l'usager.
- Réponse à une donnée d'un système ou appareil externe, e.g., traitement d'une transaction, traitement d'un événement (alarme).
- Maintenance de Routine, e.g., sauvegarde de dossier, nettoyage de la base de données.

Développer les profils opérationnels

- *Identifier qui/quoi peut entreprendre les opérations.*
  - Usagers (de différents types), systèmes externes et appareils, le système lui-même.
- *Créer une liste d'opérations pour chaque initiateur et les résultats consolidés.*
  - Source : besoins, manuels de l'usager, prototypes, version des programmes antécédents, discussion avec les usagers probables.
  - 20 à plusieurs centaines d'opérations sont typiques.
- *Déterminer le taux d'occurrence (par heure) des opérations individuelles.*
  - Champs de données existants, enregistrements des champs d'opérations, simulations, estimations.
- *Dériver les probabilités d'événements.*

Exemple : Fone Follower (FF)

- Besoins :
  - Faire suivre les appels téléphoniques entrants (voix, fax) de n'importe où.
  - Le souscripteur appelle FF, entre les numéros de téléphone où il sera joignable en fonction de temps.
  - FF fait suivre les appels entrants en provenance du réseau (voix, fax) au souscripteur. S'il n'y a pas de réponse à l'appel vocal, le souscripteur est appelé sur son téléavertisseur (si le souscripteur en a un). Si aucune réponse ou pas de téléavertisseur, les appels vocaux sont transmis au courrier vocal.
  - Le souscripteur voit le service comme un service téléphonique standard combiné avec FF.
  - FF utilise le système d'opération vendeur-approvisionnement de fiabilité inconnue.

Initiateurs des opérations

- Les événements actionneurs des systèmes ont souvent plusieurs systèmes externes qui peuvent eux-mêmes initier les opérations.
- Typiquement, le système sous étude peut initier lui-même des opérations administratives et de maintenance.
- FF:
  - Types d'usagers : souscripteurs, administrateurs de système.
  - Système externe : réseau téléphonique.
  - FF (vérifications, sauvegardes).

Liste d'opérations FF

- Souscripteur
  - Entre le numéro téléphonique
- Administrateur de système
  - Ajoute le souscripteur
  - Supprime le souscripteur
- Réseau
  - Proc. appel vocal, pas de téléavertisseur, réponse
  - Proc. appel vocal, pas de téléavertisseur, pas de réponse
  - Proc. appel vocal, téléavertisseur, réponse
  - Proc. appel vocal, téléavertisseur, réponse sur téléavertisseur
  - Proc. appel vocal, téléavertisseur, pas réponse sur téléavertisseur
  - Proc. appel télécopieur
- FF
  - Vérifie la section base de données des numéros de téléphone
  - Récupère d'une défaillance matérielle

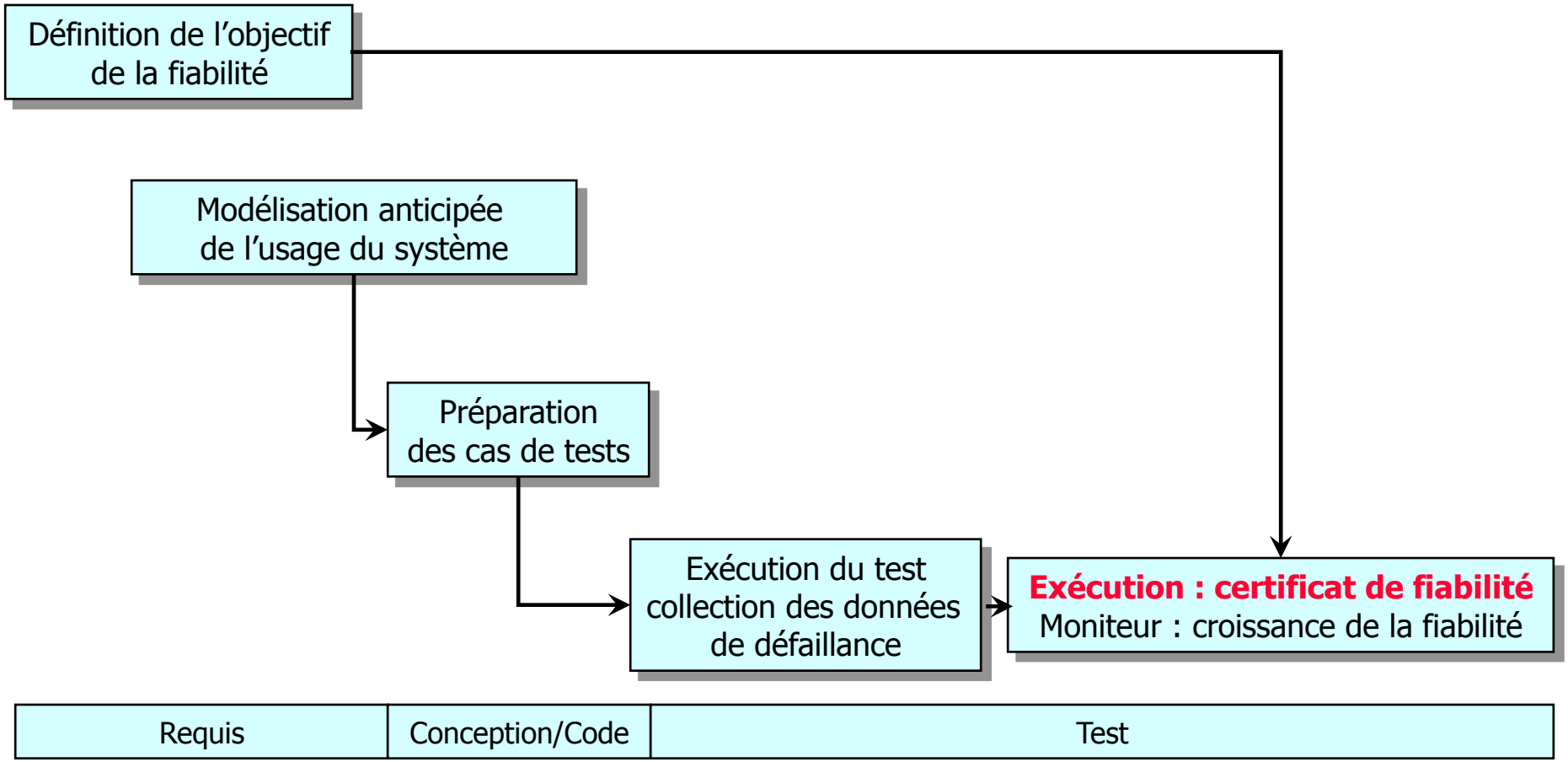
Profil opérationnel FF

Opération	**Taux d'occ. (par hre)**	**Prob.d'occ.**
Entre un numéro téléphone	10,000	0,10
Ajoute un souscripteur	50	0,0005
Supprime un souscripteur	50	0,0005
Proc. appel vocal, pas téléavertisseur, réponse	18,000	0,18
Proc. appel vocal, pas téléavertisseur, pas réponse	17,000	0,17
Proc. appel vocal, téléavertisseur, réponse	17,000	0,17
Proc. appel vocal, téléavertisseur, réponse sur téléavertisseur	12,000	0,12
Proc. appel vocal, téléavertisseur, pas réponse sur téléavertisseur	10,000	0,10
Proc. appel télécopieur	15,000	0,15
Vérifie section base de données numéro téléphone	900	0,009
Rétablit d'une défaillance matérielle	0,1	0,000001

Test statistique

- Test basé sur les profils opérationnels
- Cette forme de test a l'avantage de tester plus intensément les fonctions du système qui seront le plus utilisées.
- Bon pour estimer la fiabilité, mais pas très efficace pour trouver les défauts.
- En conséquence, nous différencions le test qui vise à trouver les défauts (vérification) et le test dont le but est l'évaluation de la fiabilité (validation).
- Certains chercheurs proposent des techniques qui combinent les tests de boîte blanche et les tests statistiques …

Processus d'ingénierie de la fiabilité du logiciel



Pouvons-nous accepter un composant ?

- *Test de certification* : Montre qu'un composant (acquis ou développé) satisfait un objectif de fiabilité donné (e.g., intensité de défaillance).
- Génère le test de données aléatoirement selon l'usage du modèle (e.g., profil opérationnel).
- Enregistre toutes les défaillances (aussi les défaillances multiples) mais *ne corrige pas.*

Procédure

- Utilise un *diagramme de contrôle de test d'hypothèse* pour montrer que l'objectif de fiabilité est/n'est pas satisfait.
  - *Charte de démonstration de fiabilité*
    - » Collectionne le temps auquel les défaillances se sont produites.
    - » Normalise les données en utilisant l'objectif de l'intensité de la défaillance (utilisant les mêmes unités!).
    - » Relève chaque défaillance dans un diagramme.
    - » Accepte ou rejette le composant selon la région dans laquelle les défaillances surviennent.

Charte de démonstration de la fiabilité

Charte de démonstration de la fiabilité

Défaillance #

Rejeter

Continuer

Accepter

Défail.#	Mappels à la défaillance	Mesure normalisée
1	0.1875	0.75
2	0.3125	1.25
3	1.25	5.0

=> accepté

Objectif de l'intensité de la défaillance :
4 défaillances/Mappels

John Musa, Software Reliability, 1998

Temps normalisé de la défaillance
(temps de défaillance × objectif de l'intensité de la défaillance

Créer des chartes de démonstration

- Sélectionner le *ratio de discrimination* γ (facteur acceptable de l'erreur dans l'estimation de l'intensité de la défaillance).
- Sélectionner le *risque de consommateur* β (probabilité d'accepter un système qui ne satisfait pas l'objectif d'intensité de la défaillance).
- Sélectionner le *risque du fournisseur* α (probabilité de rejeter un système qui satisfait l'objectif d'intensité de la défaillance).
- Paramètres recommandés par défaut (γ=2, β=0.1, α=0.1).
  - Risque de 10% d'accepter par erreur un composant quand l'intensité de défaillance est actuellement ≥ 2 * l'objectif d'intensité de la défaillance.
  - Risque de 10% de rejeter par erreur un composant quand l'intensité de défaillance est actuellement ≤ 1/2 * l'objectif d'intensité de la défaillance.

Lignes de limite

La prochaine étape est de construire des lignes de limite (entre Rejeter et Continuer, et Continuer et Accepter) selon la formule suivante :

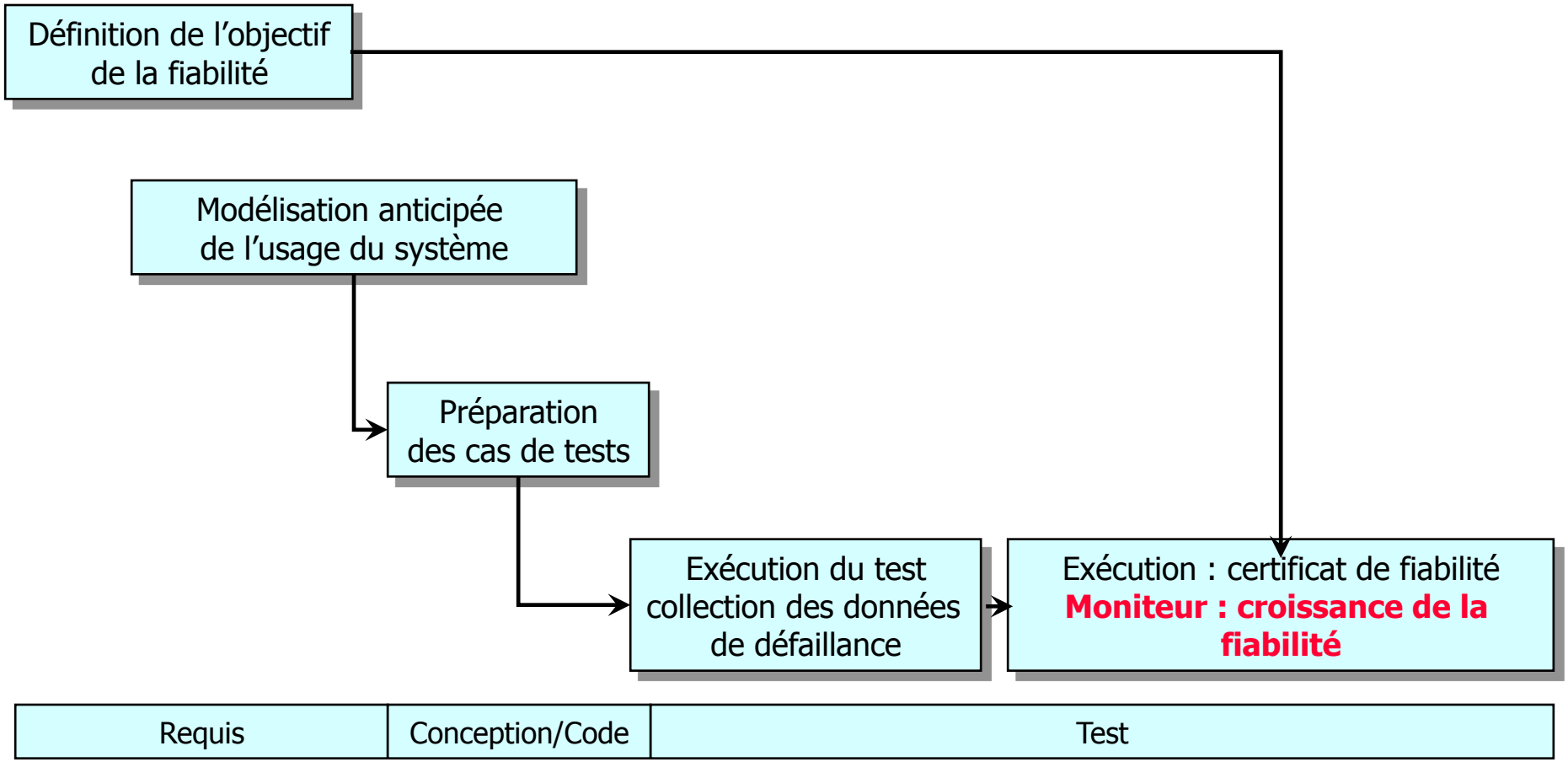
$$T_N = \frac{A - n\ln\gamma}{1-\gamma} \qquad T_N = \frac{B - n\ln\gamma}{1-\gamma} \qquad A = \ln\frac{\beta}{1-\alpha} \qquad B = \ln\frac{1-\beta}{\alpha}$$

où

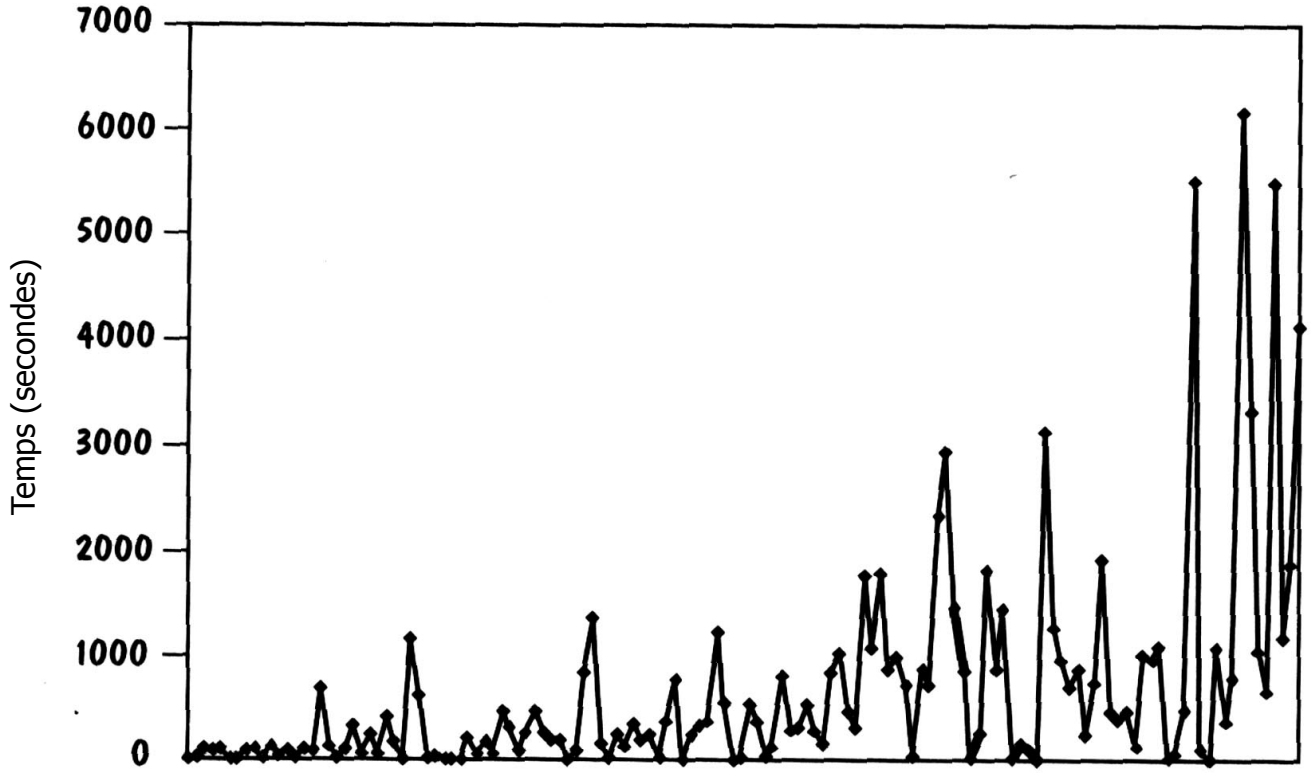
T_n = limites pour le temps normalisé de la défaillance (axe x)

n = nombre de défaillances (axe y)

Processus d'ingénierie de la fiabilité du logiciel



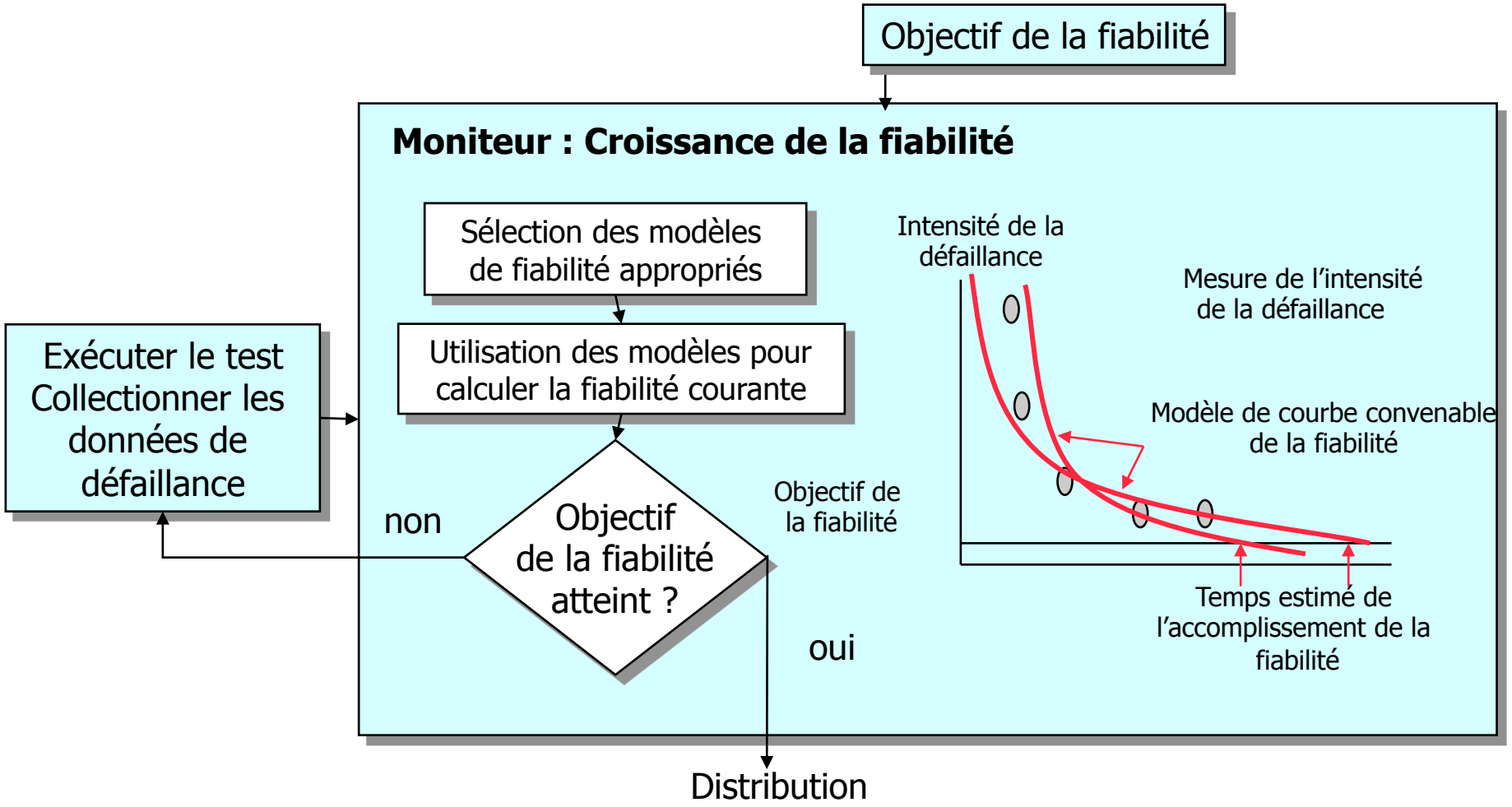
Données de défaillance : temps d'inter-défaillance



Défaillances (en ordre de production de gauche à droite)

Figure 8.7 Graphe de données de défaillances de la Table 8.3.

Pouvons-nous arrêter de tester ? si non, quand?



Modèle MUSA

- Développé par John Musa aux Laboratoires ATT Bell.
- Un des premiers proposés et encore largement appliqué.
- Il est basé sur le temps d'exécution.
- Le temps d'exécution peut être converti dans un temps calendaire (dans une seconde partie du modèle).

Hypothèses de base

- Le taux de détection des fautes est proportionnel au nombre courant de fautes dans le code.
- Le taux de détection de fautes demeure constant dans l'intervalle entre l'occurrence des fautes.
- Un faute est instantanément corrigée, avant l'introduction d'autres fautes dans le code.
- La défaillance est indépendante, quand les fautes sont détectées.
- Le logiciel sera utilisé d'une manière similaire après que les prédictions de fiabilité aient été faites.
- Chaque défaillance dans une classe de gravité a la même probabilité d'avoir été éprouvée

Hypothèses MUSA

- Hypothèses de base.
- Le nombre cumulatif de défaillance par temps t, M(t), suit un processus de Poisson avec les fonctions moyennes :

$$\mu(t) = \beta_0 \left[1 - e^{-\beta_1 t}\right] \text{with } \beta_{0,1} > 0$$

- Le temps d'exécution entre les défaillances est exponentiellement distribué i.e. le taux aléatoire est constant.

Hypothèses MUSA ...

- Les ressources disponibles (testeurs, débogueurs, programmeurs) sont constantes durant le segment de temps pendant lequel le système est observé.
- Le personnel identificateur de fautes peut être complètement utilisé et l'utilisation d'un ordinateur est constante.
- Le personnel correcteur de fautes est établi par la fixation d'une longueur maximale de la file de fautes pour n'importe quel correcteur de faute.

Données requises MUSA

- Pour appliquer le modèle MUSA, nous avons besoin
  - du temps actuel où le logiciel a fait défaut, ou
  - du temps écoulé entre les défaillances.

La forme du modèle MUSA

Soient les hypothèses :

Nombre cumulatif moyen de fautes

$$\mu(t) = \beta_0\left[1 - e^{-\beta_1 t}\right] \text{with } \beta_{0,1} > 0$$

Intensité de la défaillance

$$\lambda(t) = \beta_0\beta_1 e^{-\beta_1 t}$$

Ceci est un modèle de défaillances finies avec le nombre total de fautes :

$$\beta_0$$

Choix des paramètres MUSA

Pour que l'échantillon observé soit celui le plus susceptible de se produire parmi tous les échantillons possibles, on estime les bêtas comme suit

$$\hat{\beta}_0 = \frac{n}{1 - \exp\left[-\hat{\beta}_1 \left(t_n + x\right)\right]}$$

$$\frac{n}{\hat{\beta}_1} - \frac{n\left(t_n + x\right)}{\exp\left[\hat{\beta}_1 \left(t_n + x\right)\right] - 1} - \sum_{i=1}^{n} t_i = 0$$

Substituer les betas estimés dans l'équation du modèle pour obtenir une estimation de R(t), $\mu(t), \lambda(t), \ldots,$ etc .

Exemple

Supposons que les temps de défaillances suivants aient été observés : 10, 18, 32, 49, 64, 86, 105, 132, 167, 207 en heures et supposons un additionnel 15 CPU heures sans défaillances :

$$\hat{\beta}_0 = \frac{10}{1 - \exp\left(-222\hat{\beta}_1\right)}$$

$$\frac{10}{\hat{\beta}_1} - \frac{2220}{\exp\left[222\hat{\beta}_1\right] - 1} - 870 = 0$$

avec le résultat de : $\hat{\beta}_0 = 13.6$ et $\hat{\beta}_1 = 0.006$.

Interprétation des paramètres

En faisant la correspondance :

$$\beta_1 = \lambda_0 \, B \quad \text{et} \quad \beta_0 = \nu_0$$

Nous représentons le taux constant aléatoire λ_0

et la réduction de fautes de facteur B (e.g., 0.95).

Solution MUSA de base

Considérons l'intensité de défaillance comme fonction du nombre moyen de défaillances

$$\lambda(\mu) = \lambda_0\left(1 - \frac{\mu}{\nu_0}\right)$$

$\lambda(\mu)$: intensité de défaillance.
λ_0: intensité initiale de défaillance au départ (taux constant aléatoire)
μ: Nombre moyen total de défaillances à un moment donné
ν_0: Nombre total de défaillances sur un temps infini

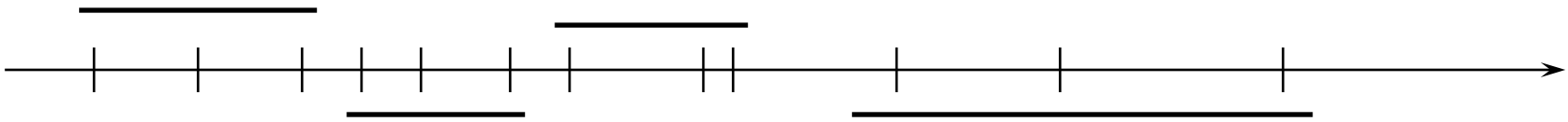
Trouver la plus petite solution de cette ligne avec les données actuelles. Nous pouvons ainsi déterminer l'effort encore nécessaire. Soit λ_F, λ_P les valeurs finale et courante de l'intensité de défaillance.

$$\Delta\mu = \frac{\nu_0}{\lambda_0}\left(\lambda_P - \lambda_F\right) \qquad \Delta t = \frac{\nu_0}{\lambda_0}\lg\left(\frac{\lambda_P}{\lambda_F}\right)$$

Groupement des données

- ***Choisissez*** un nombre de défaillances que vous souhaitez considérer pour chaque sous-intervalle disons k.
- Divisez l'ensemble des données en sous-ensembles indépendants p contenant exactement k défaillances (p est le plus petit nombre entier supérieur ou égal au nombre total de défaillances divisé par k).
- Estimer l'intensité de la défaillance comme le ratio entre le nombre de défaillances dans l'intervalle et la durée de l'intervalle.

Exemple



k=3, premier intervalle 3 défaillances, second intervalle 3, troisième 3, ...

Équations

$$r_l = \frac{k}{t_{kl} - t_{k(l-1)}} \quad l = 1,2,\ldots, p-1$$

$$r_p = \frac{m_e - k(p-1)}{t_e - t_{k(p-1)}}$$

m_e nombre total de défaillances.

t_e temps d'arrêt des observations.

$m_l = k(l-1)$ défaillance cumulative jusqu'à l'intervalle l.

Construire les couples : (r_l, m_l) en accord avec le modèle

$$\lambda(\mu) = \lambda_0\left(1 - \frac{\mu}{\nu_0}\right)$$

Solution

Pour les couples $(x_i, y_i) = (m_l, r_l)$

Si : y=bx+a

On a:

$$\hat{b} = \frac{n\sum_{1}^{n} x_i y_i - \sum_{1}^{n} x_i \sum_{1}^{n} y_i}{n\sum_{1}^{n} x_i^{\,2} - \left(\sum_{1}^{n} x_i\right)^2}$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x}$$

Les pour et les contre

- Pour
  - La fiabilité peut être spécifique.
  - L'objectif et l'estimation directe de la fiabilité.
  - La prédiction du temps pour la distribution.
- Contre
  - Le modèle utilisé peut être difficile à imaginer.
  - La sélection du modèle de croissance de la fiabilité peut être difficile.
  - La mesure de temps crucial peut être difficile dans certains contextes.
  - Non applicable aux systèmes à sécurité-critique comme très hauts besoins de fiabilité, cela prendrait des années de tests à vérifier.