Chapitre 7

Mesurer et contrôler les produits (travail)

Le contenu est adapté de Richard E. Fairley (2009):
Managing and Leading Software Projects, annotated edition,
Wiley-IEEE Computer Society

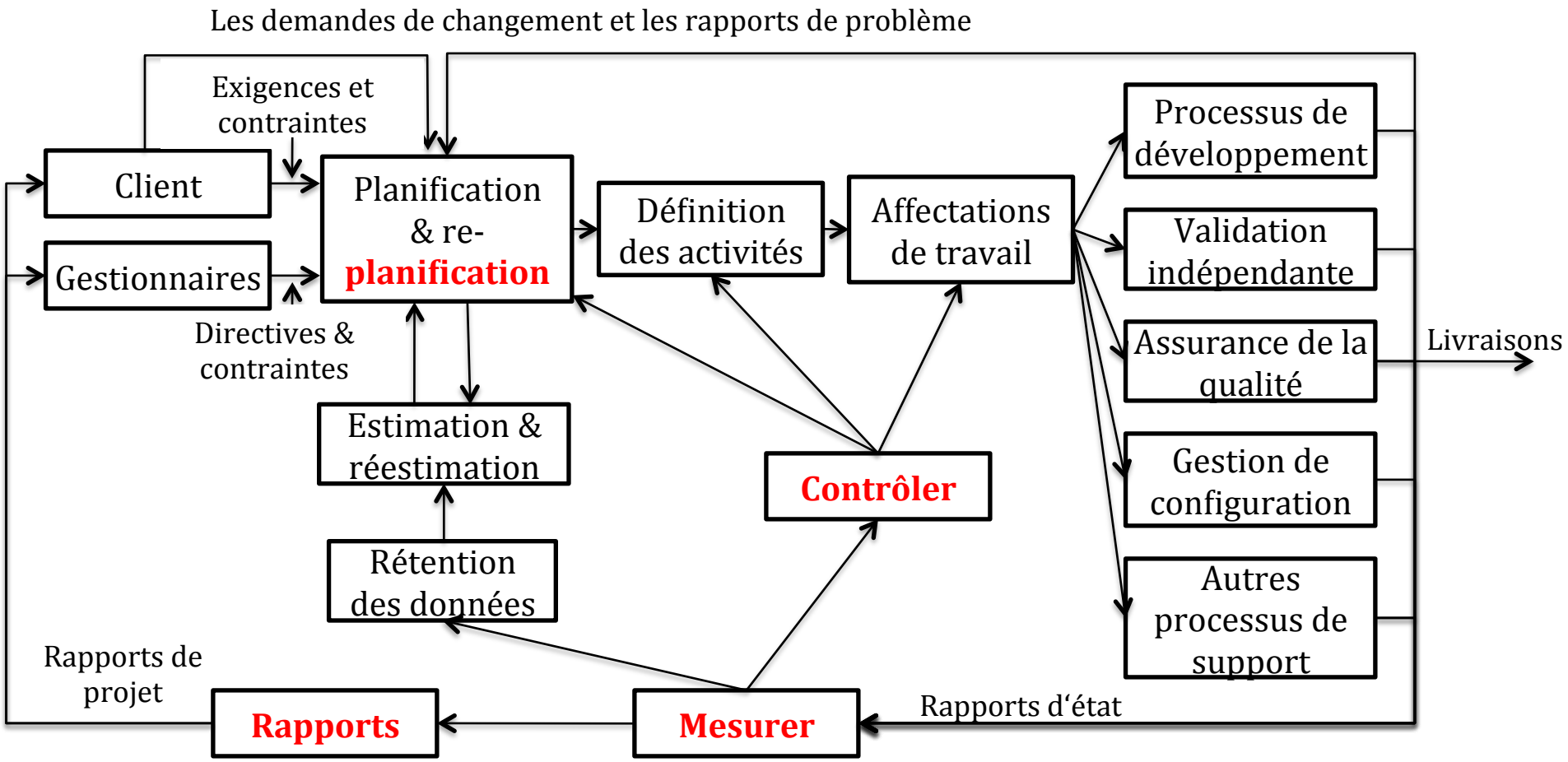
Préparé par:
Mazen El-Masri , PMP, M.Sc, ABD
http://ca.linkedin.com/in/mazenmasri
Révisé par:
Jean-Marc Desharnais, ÉTS

Objectifs

- Les mesures et les échelles de mesure
- Les mesures de différents types de produits (travail)
- Le rôle de la gestion de la configuration dans la mesure et le contrôle des produits
- Le rôle des inspections, des walkthroughs et des tests
- Les différentes mesures de la complexité des logiciels
- La mesure de la fiabilité, de la disponibilité et de la maintenabilité
- La détection des défauts et le processus de correction
- Les façons de documenter et analyser les défauts et de les corriger

Le contenu est adapté de Fairley (2009)

Modèle de gestion de projet logiciel



Adapté de Fairley (2009)

Mesurer – Pourquoi?

- Fournir des indicateurs fréquents du progrès.
- Permettre de signaler très tôt les problèmes.
- Permettre l'analyse des tendances dans le projet.
- Permettre des estimations de coût final et de date d'achèvement du projet.
- Bâtir un référentiel de données historiques de projets pour l'organisation.

Le contenu est adapté de Fairley (2009)

Mesurer – c'est quoi?

- **Mesure:** cueillette, validation et analyse des informations concernant l'état du projet
- **Une mesure directe:** une mesure déterminée suite à l'étude directe du phénomène (aussi mesure de base -ISO/IEC 15939)
  - Par exemple, la taille en point de fonction COSMIC
- **Une mesure indirecte:** une mesure dérivée d'une mesure directe (aussi une mesure dérivée – ISO/IEC 15939)
  - Par exemple, la mesure de la productivité = lignes de code par mois

Le contenu est adapté de Fairley (2009)

Contrôler – c'est quoi?

- **Contrôle:** application de mesures correctives lorsque l'état de situation actuel n'est pas conforme à l'état de situation prévu.

- **État:**
  - De la quantité et de la qualité des produits (travail)
  - Du processus de développement (échéancier, budget, ressources, facteurs de risque)

Le contenu est adapté de Fairley (2009)

Échelle de mesures

- Nominal
- Ordinal
- Intervalle
- Ratio
- Absolu

Le contenu est adapté de Fairley (2009)

Exercice: utilisation des mesures

- Quelle est l'erreur la plus commune dans l'utilisation d'une mesure?

Le contenu est adapté de Fairley (2009)

Exercice: erreur la plus commune

- Additionner des valeurs qui n'ont pas de rapport entre elles? Pourquoi?
- Exemple: complexité de McCabe. On n'a pas défini l'unité de mesure.
- Autre erreur: ne pas tenir compte de l'échelle des mesures
- Exemple: facteurs d'ajustement IFPUG

Mesurer (et contrôler) quoi?

- Les attributs du produit: les exigences développées qui ont démontré leur capacité
- Les attributs de la qualité du produit: les défauts, la fiabilité, la disponibilité, le temps de réponse, le débit…(ISO/IEC 25010)
- L'effort: la quantité de travail dépensée pour les diverses activités de travail
- L'échéancier: la réalisation de jalons (étapes)
- Le coût: les dépenses pour différents types de ressources, y compris l'effort
- Le progrès: les produits de travail qui sont complétés, acceptés et validés (par version)
- Le risque: l'état des facteurs de risque et les activités de mitigation

Le contenu est adapté de Fairley (2009)

Exercice: mesure de base et mesure dérivée

- L'emploi de mesure directe et indirecte, puis de mesure de base et mesure dérivée est souvent considéré comme synonyme.
- Qu'en pensez-vous?
- Quelle différence y a t'il entre mesure directe et indirecte et entre mesure de base et mesure dérivée?

Le contenu est adapté de Fairley (2009)

Certaines mesures de base

Mesures	Mesures de base
Taille	Points de fonction
Nombre de personnes	Nombre de programmeurs et de testeurs
Progrès	Nombre d'exigences validées et acceptées (baselined) Nombre d'exigences modifiées
Utilisation des ressources	Cycles du processeur utilisé Octets de mémoire utilisés
Temps	Nombre de semaines jusqu‘au jalon Temps de réponse de l’ordinateur
Qualité	Nombre de défauts Temps de réponse de l’ordinateur

Le contenu est adapté de Fairley (2009)

Certaines mesures dérivées

Mesures	Mesures dérivées
Taille	Heures par points de fonction
Productivité	Heures par lignes de code
Taux de testing	Nombre de tests effectués / personnel-jour
Densité de défauts	Nombre de défauts / KLOC (K = mille)
Efficacité de détection	Nombre de défauts détectés / personnel-heure
Stabilité des exigences	Nombre actuel / nombre initial
Performance des coûts	Coût réel / montant budgétisé

Le contenu est adapté de Fairley (2009)

Mesure et Contrôle– Les attributs de produit

- La qualité et la quantité des exigences du logiciel
- La taille du logiciel
- La complexité du logiciel
- La qualité du logiciel

Le contenu est adapté de Fairley (2009)

Mesure et Contrôle– Les attributs de produit

- **La qualité et la quantité des exigences du logiciel**
- La taille du logiciel
- La complexité du logiciel
- La qualité du logiciel

Le contenu est adapté de Fairley (2009)

Mesure – La quantité des exigences du logiciel

- Compter le nombre de caractéristiques opérationnelles essentielles (et des attributs de qualité) complétées
- Compter le nombre d'autres caractéristiques opérationnelles (non essentielles) complétées
- Compter le nombre de points de fonction
- Compter le nombre de cas d'utilisation
- Compter le nombre de scénarios opérationnels
- Compter les facteurs spécifiques au domaine
  - Entrées, sorties, fichiers…
  - Fenêtres, menus…
  - Interruptions, signaux de contrôle

Le contenu est adapté de Fairley (2009)

Technique de vérification de la quantité des exigences du système

- Les exigences du système doivent être suffisamment précises et détaillées.
- Cependant, certaines exigences peuvent être moins précises ou moins détaillées que d'autres. Pour cela, chacune des exigences doit être évaluée à partir d'une échelle (faible, modéré, élevé) de:
  - Niveau de détail
  - Précision
  - Adéquation (pour la planification des tests)
- Une cote moyenne sur l'échelle de quantification indique que la quantification est incomplète ou impraticable.

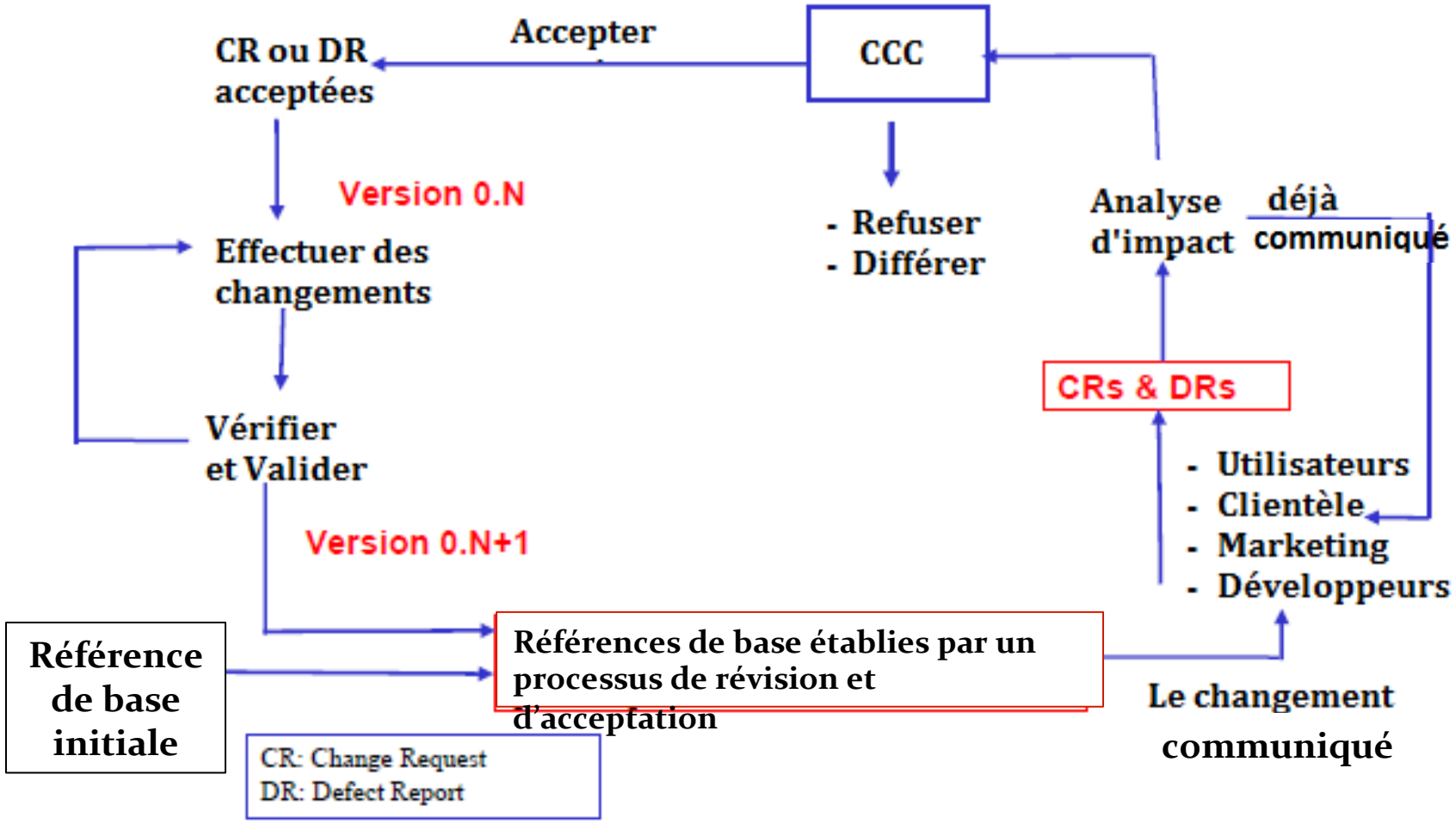
Le contenu est adapté de Fairley (2009)

Contrôle – Les exigences du logiciel

- Le contrôle des produits de travail exige un processus de contrôle des changements.
- Les obligations d'un comité de contrôle des changements comprennent:
  - Examiner les CR (change request), DR (defect report) et les rapports d'analyse d'impact
  - Accepter, refuser ou différer les CR et les DR
  - Suivre le progrès des CR et DR acceptés jusqu‘à la clôture

Le contenu est adapté de Fairley (2009)

Contrôle – Les exigences du logiciel



Le contenu est adapté de Fairley (2009)

Gabarit pour demande de changement

Numéro de la demande de changement :

Soumis par:

Date de soumission :

Décision:

__accepter ___différer ___refuser ___un duplicata

Priorité (si accepté) :

___haute ___moyenne ___basse

Références de base ajoutées (nom(s) et numéro(s) de version) :

Références de base modifiées (nom(s) et numéro(s) de version) :

Personnel-heure nécessaire au changement :

Date d'approbation:

Acceptation par (sign-off) :

Date de clôture :

Personnel avisé du changement :

Le contenu est adapté de Fairley (2009)

Mesure – La qualité des exigences du logiciel

- **Vérification:** processus de détermination de la mesure dans laquelle les exigences satisfont aux conditions imposées par d'autres produits et/ou processus de travail.
  - Est-ce que les exigences opérationnelles sont complètes, correctes et cohérentes par rapport aux besoins des utilisateurs, aux attentes du client et aux conditions de l'acquéreur?
  - Est-ce que les spécifications techniques sont exactes, complètes et cohérentes par rapport aux besoins opérationnels?
  - Est-ce que le code des programmeurs est exact, complet et cohérent par rapport aux spécifications techniques?

Le contenu est adapté de Fairley (2009)

Technique de vérification des exigences opérationnelles

- Par des matrices de traçabilité

Exemple:

- Une matrice pour valider les exigences du système avec les spécifications techniques.
- Une matrice de traçabilité des exigences à la source. Ex:

Source → / Exigence ↓	Client	Groupe d'utilisateur #1	Groupe d'utilisateur #2	Marketing
[1]	X	X		
[2]		X	X	X
[3]				X
[4]		X		

Le contenu est adapté de Fairley (2009)

Techniques de vérification des spécifications techniques

- Traçabilité: matrice pour valider les spécifications techniques aux scénarios de test.
- Walkthroughs: pour consulter les produits de travail et examiner et communiquer les questions ou les problèmes techniques.
  - Aucune documentation, aucun rôle assigné, aucune analyse de résultats
  - Réunion technique avec les développeurs
  - Objectif principal = communiquer les problèmes techniques

Le contenu est adapté de Fairley (2009)

Techniques importantes de vérification du code

- **Inspections de code**: pour trouver et documenter les défauts ainsi que les processus de correction.
- Le processus d'inspection exige que le gestionnaire de projet prévoit suffisamment de temps pour la préparation de l'inspection, la tenue de la réunion d'inspection, les reprises, et les activités de suivi des inspections.
- Mesure: le nombre d'inspections de code effectué par rapport au nombre prévu pendant le cycle de travail visé.

Le contenu est adapté de Fairley (2009)

Techniques importantes de vérification du code : Un gabarit de rapport de défauts

Numéro du défaut:

Proposé par:

Date de soumission:

Brève description de la défaillance:

Le niveau de gravité: __ Majeur___ Mineur___ Inconvénient

Priorité pour la fixation: __Immédiatement___ Sitôt `dés possible ___Différer

Phase durant laquelle l'erreur a été commise:

__ Conception __Dév. exigences __Mise en œuvre __ Vérification __ Validation

Phase dans laquelle l'erreur a été trouvée:

__ Conception __Dév. exigences __Mise en œuvre __ Vérification

__ Validation__ Opérations

Le contenu est adapté de Fairley (2009)

Techniques importantes de vérification du code: Un gabarit de rapport de défauts (suite)

Type d'erreur:

__ Incomplet ___ incorrect ___ incohérent __Autre (Préciser): ____

Comment l'erreur a été détectée:

__ Inspection __ Test__ Démo. __ Autre (préciser):_____

Base(s) de référence modifiée(s) pour corriger l'erreur (nom(s) & numéro(s) de la version):

Personnel - heure requis pour la correction:

Date d'approbation:

Approbation par (sign-off):

Date de clôture:

Personnel avisé des changements:

Mesure et Contrôle– Les attributs de produit

- La qualité et la quantité des exigences du logiciel
- **La taille du logiciel (voir chapitre 6)**
- La complexité du logiciel
- La qualité du logiciel

Mesure et Contrôle– Les attributs de produit

- La qualité et la quantité des exigences du logiciel
- La taille du logiciel
- **La complexité du logiciel**
- La qualité du logiciel

Le contenu est adapté de Fairley (2009)

Mesure – La complexité du logiciel

Facteur d'ajustement de la complexité (COCOMO)

- La complexité des opérations de contrôle (e.g.: if-then-else)
- La complexité des calculs
- La complexité des opérations de gestion des données
- La complexité des systèmes d'exploitation
- La complexité de l'interface utilisateur

(0.7 <= FAC <= 1.65)

Mesure et Contrôle– Les caractéristiques de produit

- La qualité et la quantité des exigences du logiciel
- La taille du logiciel
- La complexité du logiciel
- La qualité du logiciel

Le contenu est adapté de Fairley (2009)

Exercice: mesurer la qualité

- La norme ISO 9126 vise à mesurer la qualité du logiciel.
- Quelle est la principale faiblesse de cette norme?

Le contenu est adapté de Fairley (2009)

Attributs de qualité du logiciel

Les attributs de qualité requis pour un logiciel incluent :

- **La sûreté** : ne cause pas de tort
- **La sécurité** : protège les données (et l'accès)
- **La fiabilité :** n'échoue pas plus de prévu
- **La disponibilité** : est rapidement réparé après échec
- **L'utilisabilité :** peut être utilisé efficacement
- **La maintenabilité** : est facile à modifier

Différents systèmes ont différents types et différents niveaux d'attributs de qualité

Le contenu est adapté de Fairley (2009)

Les échecs, défauts (bogues) et erreurs

- Un échec se produit lorsque le logiciel ne répond pas aux exigences, besoins ou attentes alors qu'il fonctionne dans les conditions et l'environnement prévus.
- Un échec est le résultat d'un ou de plusieurs défauts.
- Les défauts sont le résultat d'erreurs humaines.
- Un défaut est quelque chose qui rend le client ou l'utilisateur malheureux (définition de six sigma)
- Les défauts rencontrés lors des opérations produisent des défaillances (échecs) qui entraînent un système à effectuer des opérations incorrectes ou à se comporter d'une manière inattendue.

Le contenu est adapté de Fairley (2009)

Les échecs, défauts (bogues) et erreurs

- Certains niveaux de gravité des échecs:
  - **Catastrophique** : Entraîne la perte de plusieurs vies et/ou une perte significative de propriété ou d'informations.
  - **Critique**: Entraîne la perte d'une vie et/ou grave perte de propriété ou d'informations.
  - **Sévère**: Entraîne des blessures graves à une ou plusieurs personnes ou une perte de propriété ou d'informations.
  - **Marginal**: Entraîne des blessures mineures à une ou plusieurs personnes et/ou une dégradation des performances et/ou une perte temporaire de disponibilité du système.
  - **Mineur**: Entraîne en maintenance non programmée.

Le contenu est adapté de Fairley (2009)

Les échecs, défauts (bogues) et erreurs:
Pourquoi les hommes font des erreurs?

- Les ruptures dans la communication et la coordination sont les raisons principales des erreurs de génie logiciel:
  - L'information nécessaire n'est pas obtenue
  - L'information n'est plus à jour
  - L'information a changé
  - L'information est mal interprétée
  - Les rôles et responsabilités sont mal définis ou incompris
- Le manque de formation, d'outils ou d'expérience

Le contenu est adapté de Fairley (2009)

Les échecs, défauts (bogues) et erreurs:
Les types de reprises

- Les défauts observés au cours du développement de logiciels engendre des reprises qui pourraient être évitées.
- Il y a trois types de reprise:
  1. **Évolutive**: apporte une valeur ajoutée au produit évolutif et doit être accompagnée par des ajustements à l'échéancier et aux ressources, si nécessaire.
  2. **Rétrospective**: reprend le travail qui aurait dû être fait auparavant
  3. **Corrective**: corrige les défauts

Le contenu est adapté de Fairley (2009)

Les échecs, défauts (bogues) et erreurs:
Un gabarit de lots de travail de reprises

- Numéro du lots de travail original:
- Type de reprise:

  EV: évolutive RE: rétrospective CO: corrective
- Description de la reprise:
- Date de début:
- Date de fin:
- Efforts consacrés:
- Autres produits de travail modifiés:
- Nouveaux critères d'acceptation du produit de travail:
- En cas de reprise corrective:
  - Type de défaut (exigences, conception, code,..)
  - Phase de découverte du défaut
  - Brève description de la cause

Les échecs, défauts (bogues) et erreurs:
Le rôle de la V&V et du processus d'amélioration

- Un défaut qui produit un échec entraîne une perte de qualité du produit qui peut se manifester par le manque de:

  (1) sureté, (2) sécurité, (3) fiabilité, (4) disponibilité, (5) convivialité ou (6) maintenabilité.

- La détection et la prévention des défauts sont donc des mécanismes sous-jacents du contrôle de la qualité logicielle
  - Le processus vérification et validation (V & V) sert de mécanisme de détection des défauts.
  - Le processus d'amélioration sert de mécanisme de prévention des défauts.

Le contenu est adapté de Fairley (2009)

Suivi des défauts par phase de travail

Phase / Type	Exigences	Design	Code	Test	Opération	Total
Exigences	DEe	DEd	DEc	DEt	DEo	DET
Design	_	DDd	DDc	DDt	Ddo	DDT
Code	_	_	DCc	DCt	DCo	DCT
Test	_	_	_	DTt	DTo	DTT
Total	ESt	DSt	CSt	TSt	OSt	**TOTAL**

- % des défauts d'exigences détectés durant la phase de développement des exigences **= (DEe x 100)/DET**
- % des défauts de design qui passent la phase du design = [1 – (DDd/ DDt)] x 100
- % total de défauts détectés par les utilisateurs : OSt x 100/TOTAL

Les échecs, défauts (bogues) et erreurs:
Confinement des défauts (defect containment)

- Le confinement (containment) des défauts est calculé par le pourcentage de défauts détectés pendant l'activité de travail qui génère le produit du travail.
- Appelé aussi «l'efficacité de détection des défauts ».
- L'objectif de confinement de défauts devrait être de détecter > 75% défauts à chaque étape de travail.

Le contenu est adapté de Fairley (2009)

Mesurer la fiabilité, la disponibilité et la maintenabilité

- **Fiabilité:** mesurée par la probabilité qu'un système fonctionne sans défaillance dans des conditions énoncées pour une période de temps précise.
  - La fiabilité est mesurée par le temps moyen de fonctionnement entre défaillances (MTTF – mean time to failure)
  - MTTF: quantité moyenne de temps où le système est opérationnel entre chacune des pannes.

Le contenu est adapté de Fairley (2009)

Mesurer la fiabilité, la disponibilité et la maintenabilité

- **Disponibilité:** la probabilité qu'un système soit opérationnel à un moment précis dans le temps.
  - D = MTTF/(MTTF +MTTR)
  - MTTR: Délai moyen pour la réparation (mean time to repair)

    Moyenne de temps nécessaire pour redémarrer un système ayant échoué.

- **Maintenabilité:** la probabilité qu'une activité de maintenance puisse être accomplie dans un délai (T) précis à l'aide des techniques, outils et procédures spécifiés.
  - M = T /MTTR