



CQPM Automaticien – 2024/2025

MODULE R2 – REGLES DE QUALIFICATION D'UN LOGICIEL



01.

Présentation

02.

Rappel du cycle en V

03.

Qualification d'un logiciel

04.

Gestion des non-conformités

05.

Gestion des modifications
logiciels

06.

Analyse de non-régression

07.

Gestion de configuration

08.

Règles de conception et de
réalisation des logiciels
d'automatisme

08h30 – 12h00

Pause de 15 min vers 10h30

13h00 – 16h30

Pause de 15 min vers 15h00

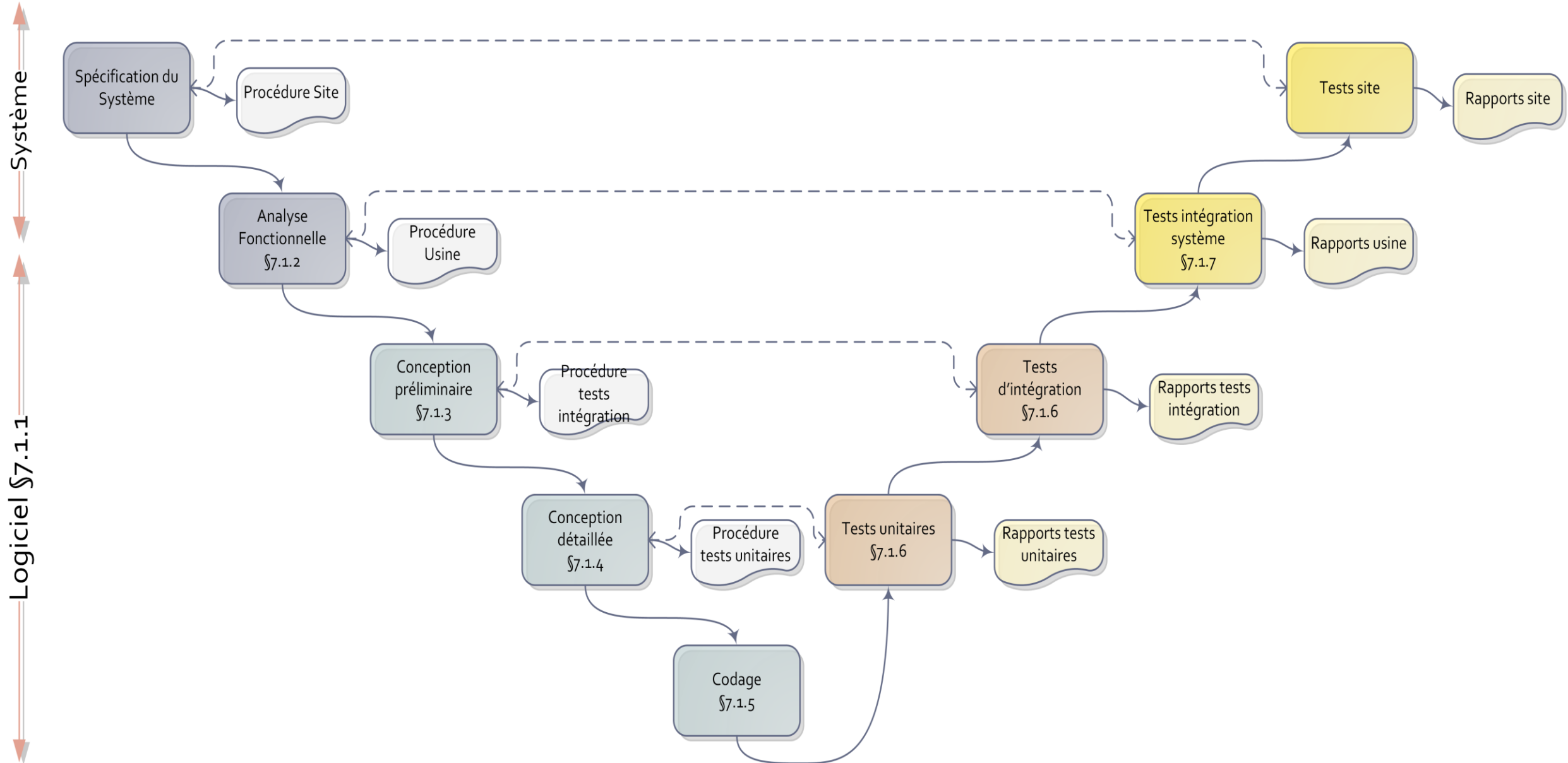
- Nom/Prénom
- Société
- Cours
- Attente de la formation

Exercice d'introduction à la formation

Un treuil de levage de 5t est ajouté sur un pont.

Définir les documents qui vont vous permettre de réaliser et valider cette modification

Cycle en V



Qualification d'un logiciel – Tests unitaires

Les tests unitaires sont une phase des tests de qualification d'un logiciel et permettent de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée "unité" ou "module").

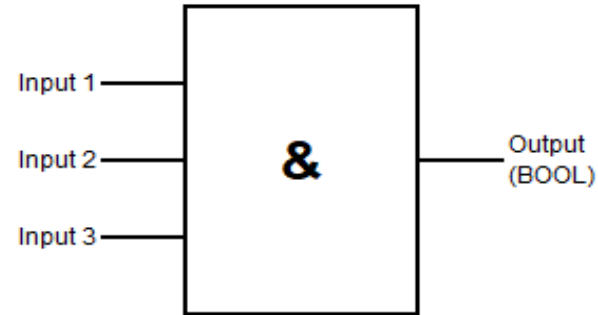
Il s'agit pour le programmeur de tester un module, indépendamment du reste du programme, ceci afin de s'assurer qu'il réponde aux spécifications fonctionnelles et qu'il fonctionne correctement en toutes circonstances.

Les tests unitaires doivent couvrir toute la plage de valeur de ou des entrée(s) du module et doivent comporter toutes les combinaisons des différentes entrées.

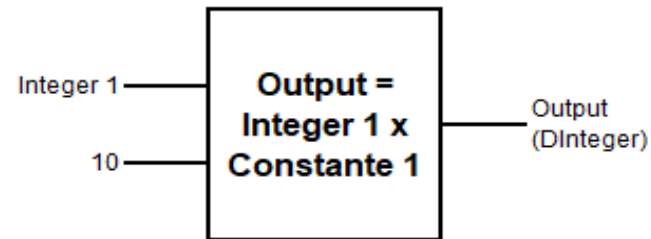
(les modules fournisseurs ne sont pas à tester en tests unitaires. Ces tests ont été réalisés par les fournisseurs).

Exemples:

Qualification d'un logiciel – Tests unitaires



Valeur de l'entrée "Input 1"	Valeur de l'entrée "Input 2"	Valeur de l'entrée "Input 3"	Valeur attendue de la sortie "Output"	Valeur relevée de la sortie "Output"
0	0	0	0	
0	1	0	0	
0	0	1	0	
0	1	1	0	
1	0	0	0	
1	1	0	0	
1	0	1	0	
1	1	1	1	



Valeur de l'entrée "Constante 1"	Valeur de l'entrée "Integer 1"	Valeur attendue de la sortie "Output"	Valeur relevée de la sortie "Output"
10	-32768	-327680	
10	-30000	-300000	
10	-25000	-250000	
10	-20000	-200000	
10	-15000	-150000	
10	-10000	-100000	
10	-5000	-50000	
10	0	0	
10	5000	50000	
10	10000	100000	
10	15000	150000	
10	20000	200000	
10	25000	250000	
10	30000	300000	
10	32767	327670	

Qualification d'un logiciel – Tests d'intégration

Les tests d'intégration sont une phase des tests de qualification d'un logiciel. Ils sont précédés par les tests unitaires et généralement suivis par les tests d'intégration du système. Ils vérifient le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme. Dans les tests d'intégration, chacun des modules indépendants du logiciel sont assemblés et testés dans l'ensemble.

L'objectif de chaque phase de test est de détecter les erreurs qui n'ont pas pu être détectées lors de la précédente phase. Pour cela, les tests d'intégration ont pour cible de détecter les erreurs non détectables par les tests unitaires.

Les tests doivent montrer que :

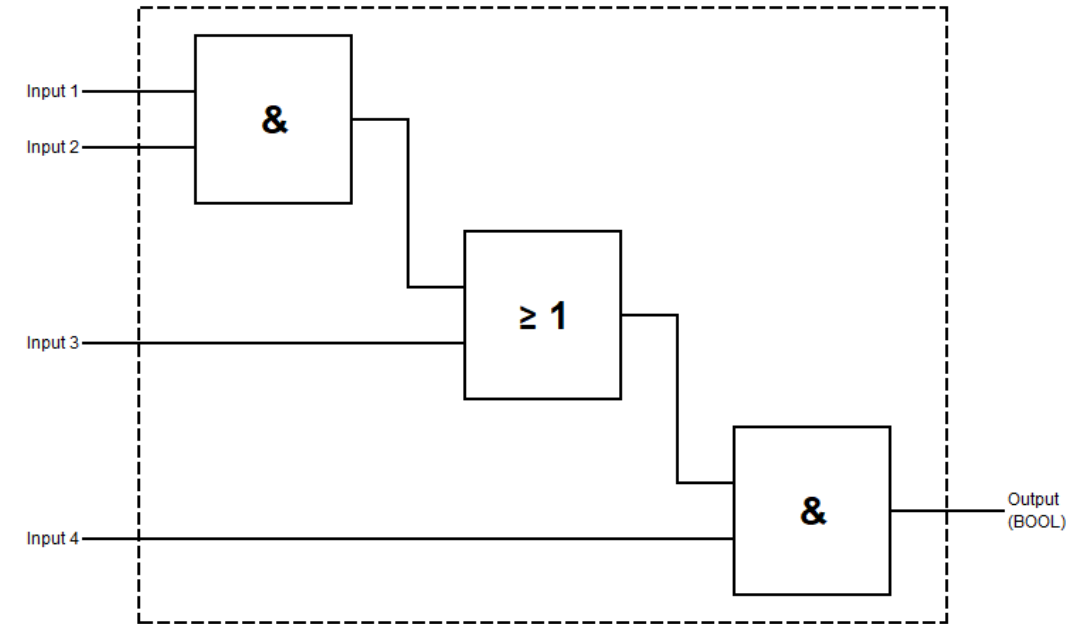
- Les séquences attendues d'événements conduisent aux fonctions souhaitées,
- Les séquences inattendues d'événements n'ont pas d'effet,
- Les éléments associés au temps, tels que les temporisations, les timeouts, les délais de réaction, sont respectés.

Les tests d'intégration permettent également de vérifier l'aspect fonctionnel, les performances et la fiabilité du logiciel.

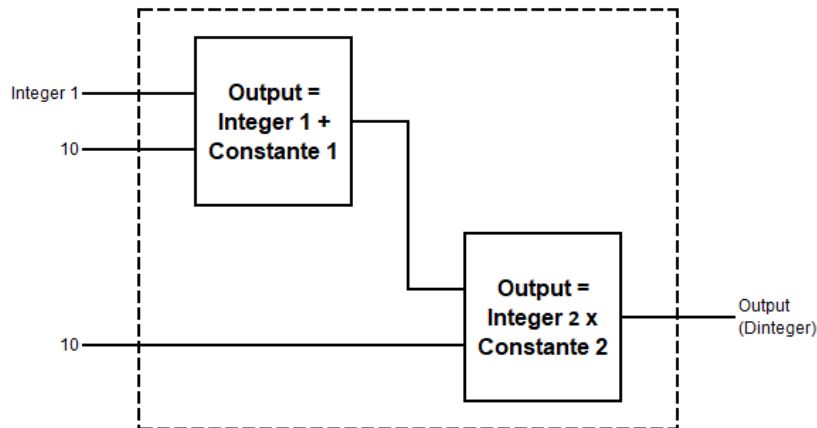
Les tests d'intégration doivent couvrir toute la plage de valeur de ou des entrée(s) du module et doivent comporter toutes les combinaisons des différentes entrées.

Exemple:

Qualification d'un logiciel – Tests d'intégration



Valeur de l'entrée "Input 1"	Valeur de l'entrée "Input 2"	Valeur de l'entrée "Input 3"	Valeur de l'entrée "Input 4"	Valeur attendue de la sortie "Output"	Valeur relevée de la sortie "Output"
0	0	0	0	0	
0	0	1	0	0	
0	0	0	1	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	1	0	0	
0	1	0	1	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	1	0	0	
1	0	0	1	0	
1	0	1	1	1	
1	1	0	0	1	
1	1	1	0	1	
1	1	0	1	1	
1	1	1	1	1	



Valeur de l'entrée "Integer 1"	Valeur de l'entrée "Constante 1"	Valeur de l'entrée "Constante 2"	Valeur attendue de la sortie "Output"	Valeur relevée de la sortie "Output"
-32768	10	10	-327780	
-30000	10	10	-300100	
-25000	10	10	-250100	
-20000	10	10	-200100	
-15000	10	10	-150100	
-10000	10	10	-100100	
-5000	10	10	-50100	
0	10	10	100	
5000	10	10	50100	
10000	10	10	100100	
15000	10	10	150100	
20000	10	10	200100	
25000	10	10	250100	
30000	10	10	300100	
32767	10	10	327770	

Qualification d'un logiciel – Tests d'intégration système

Les tests d'intégration système aussi appelés tests fonctionnels sont une phase des tests de qualification d'un logiciel. Ils sont précédés par les tests d'intégration et généralement suivis par les tests site. Ils se déroulent, généralement, en usine.

Les tests d'intégration système ont pour but de garantir que le fonctionnement constaté est identique à celui attendu dans des situations réelles d'utilisation. Très souvent, les tests d'intégration systèmes sont appliqués pour vérifier des scénarios de tests ou des schémas d'utilisation de bout en bout.

Exemple:

Qualification d'un logiciel – Tests d'intégration système

- | | |
|--|---|
| <ul style="list-style-type: none">• Check that the emergency stop safety module +2001AR-2001XK- is activated• Check that the relay +2001AR-2002XK- is activated• Check that the relay +2001AR-2003XK- is activated• Push on the power cabinet emergency stop +2001AR-2007AU-• Check that the emergency stop safety module +2001AR-2001XK- is disabled• Check that the relay +2001AR-2002XK- is disabled• Check that the relay +2001AR-2003XK- is disabled• Check the activation of the operational fault n° 317 according to the stage 1 of § 8.1.2.1.2• Check that the operational fault n° 317 has been into account according to the stage 2 of § 8.1.2.1.2• Check that the emergency stop safety module +2001AR-2001XK- is disabled• Check that the relay +2001AR-2002XK- is disabled• Check that the relay +2001AR-2003XK- is disabled• Unlock the power cabinet emergency stop +2001AR-2007AU-• Check that the emergency stop safety module +2001AR-2001XK- is activated• Check that the relay +2001AR-2002XK- is activated• Check that the relay +2001AR-2003XK- is activated• Check the erasing of the operational fault n° 317 according to the stage 3 of § 8.1.2.1.2• Check that the emergency stop safety module +2001AR-2001XK- is activated• Check that the relay +2001AR-2002XK- is activated• Check that the relay +2001AR-2003XK- is activated | <ul style="list-style-type: none"><input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done
<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done<input type="checkbox"/> Operation done |
|--|---|
- Complete APPENDIX 3 – ALARM TESTS – N° 317

Qualification d'un logiciel – Tests site

Les tests site sont une phase des tests de qualification d'un logiciel. Ils sont généralement précédés par les tests d'intégration système. Ils se déroulent sur site avec le pont ou la machine dans son environnement.

Les tests site sont identiques ou proches des tests d'intégration système.

Qualification d'un logiciel – Gestion des non-conformités

Une non-conformité est la non-satisfaction d'une exigence, un écart par rapport à une réglementation, un standard, une norme, un cahier des charges, un contrat, un document, une procédure d'essais, etc.

Elle indique donc que quelque chose a été défaillant dans un processus. Il est donc important de détecter et de corriger les non-conformités de manière efficace afin de minimiser les risques et d'améliorer la qualité et la sécurité des produits et des systèmes.

Par conséquent, il est nécessaire de déclarer facilement une non-conformité afin de pouvoir traiter la non-conformité à l'aide d'actions correctives et préventives.

8.3 Maîtrise du produit non conforme

ISO 9001:2000, Systèmes de management de la qualité — Exigences

8.3 Maîtrise du produit non conforme

L'organisme doit assurer que le produit qui n'est pas conforme aux exigences relatives au produit est identifié et maîtrisé de manière à empêcher son utilisation ou fourniture non intentionnelle. Les contrôles ainsi que les responsabilités et autorités associées pour le traitement des produits non conformes doivent être définies dans une procédure documentée.

L'organisme doit traiter le produit non conforme de l'une ou plusieurs des manières suivantes:

- a) en menant les actions permettant d'éliminer la non-conformité détectée;
- b) en autorisant son utilisation, sa libération ou son acceptation par dérogation accordée par une autorité compétente ou, le cas échéant, par le client;
- c) en menant les actions permettant d'empêcher son utilisation ou son application prévue à l'origine.

Les enregistrements de la nature des non-conformités et de toutes actions ultérieures entreprises, y compris les dérogations obtenues, doivent être conservés (voir 4.2.4).

Lorsqu'un produit non conforme est corrigé, il doit être vérifié de nouveau pour démontrer la conformité aux exigences.

Lorsqu'un produit non conforme est détecté après livraison ou après que son utilisation a commencé, l'organisme doit mener les actions adaptées aux effets, réels ou potentiels, de la non-conformité.

Qualification d'un logiciel – Gestion des modifications

Toute modification du logiciel doit être tracée dès lors que les tests unitaires commencent. Il est donc nécessaire de réaliser une phase de "débogage" importante entre la fin du codage et le début des essais unitaires.

Le processus de suivi des modifications permet de mémoriser les états des modifications tout au long du projet. Le suivi des modifications fait référence à la gestion de la configuration afin de relier les versions des logiciels et les anomalies, écarts et évolutions.

Il est indispensable d'assurer la fiabilité et la traçabilité des modifications par un outil collaboratif de gestion du changement du référentiel commun.

Un test de régression ou test de non-régression est un test ayant pour but de détecter les régressions introduites dans un logiciel après un changement effectué dans celui-ci.

Une régression est un défaut qui se produit après une modification d'un logiciel lorsque des fonctionnalités existantes ne sont plus réalisées aussi bien qu'avant. Les régressions peuvent apparaître à l'occasion de n'importe quel changement établi dans un logiciel:

- Corrections de bogues,
- Ajout de nouvelles fonctionnalités,
- Modification de fonctionnalités existantes,
- Modification d'un composant externe au logiciel lui-même (nouvelle version du système, de l'interface graphique, d'un compilateur ou d'une bibliothèque tierce qui interviennent dans son fonctionnement).

Il est donc intéressant d'être en mesure de repérer les régressions, par le biais de tests automatisés ou non, avant la mise à disposition de la nouvelle version du logiciel auprès des utilisateurs.

La gestion des évolutions fonctionnelles ou des corrections sur un logiciel existant repose sur une étape d'analyse de non-régression.

Les exigences suivantes sont appliquées:

- Identification des états avant et après modification,
- Repérage précis et exhaustif des modifications du code source,
- Exécution des programmes de test,
- Expertise des fonctions qui ne sont pas couvertes par les programmes de test et mise en place des tests nécessaires.

Qualification d'un logiciel – Gestion de configuration

La gestion de configuration est une discipline de management de projet qui permet de définir, d'identifier, de gérer et de contrôler les articles de configuration tout au long du cycle de développement d'un logiciel.

La gestion de la configuration est impérative et repose sur l'enregistrement:

- Des états des logiciels produits au cours du cycle de développement,
- Des versions des outils de développement,
- De la configuration matérielle des équipements.

Chaque état d'un logiciel produit doit être identifié puis enregistré via l'outil de suivi des modifications par l'équipe de développement.

De plus, les caractéristiques des outils de développement et de la configuration matérielle sont enregistrées dans le dossier de conception.

Ces règles et principes sont motivés par la volonté:

- D'harmoniser les solutions d'organisation programme et de programmation,
- De lister les bonnes pratiques,
- De réduire les temps de codage,
- De garantir le comportement dynamique des logiciels (performance, fiabilité,...),
- De réduire les erreurs de saisie et les oublis,
- D'atteindre un niveau de lisibilité,
- De favoriser la portabilité et la standardisation de tout ou partie des applications,
- De réduire les temps de mise en service,
- D'aider la maintenance.



www.aemco.reel.fr