

Bureau d'études ISI M1
-
Plan de validation du logiciel
APES2

HISTORIQUE DES MODIFICATIONS

Référence	[PVL]			
Version	Objet de la modification	Date	Auteur(s)	Relecteur
7.0	Finalisation du document.	11/03/04	Nicolas Delmas	Delphine De Rosso
6.0	Ajout des dernières exigences prises en compte à la fin de la phase de construction	02/03/04	Nicolas Delmas	Delphine De Rosso
5.0	Suppression et ajout d'exigences car le modèle des cas d'utilisation a été modifié.	11/02/04	Nicolas Delmas	Delphine De Rosso
4.0	Création du document	20/01/04	Nicolas Delmas	Delphine De Rosso

Sommaire

1.	Introduction	3
1.1.	Avant propos.	3
1.2.	Objet	3
1.3.	Responsabilités	3
2.	Documents applicables et de référence	4
2.1.	Documents applicables	4
2.2.	Documents de référence	4
3.	Terminologie	5
3.1.	Glossaire	5
3.2.	Acronymes et abréviations	5
4.	Exigences de test	6
4.1.	Exigences communes	6
4.2.	Exigences spécifiques	6
5.	Description du logiciel	8
5.1.	Domaine d'application et objectifs	8
5.2.	Information	8
5.2.1.	Programmes	8
5.2.2.	Documentation Utilisateur	8
5.2.3.	Documentation technique	8
5.2.4.	Code Source	9
5.3.	Découpage en composants	9
6.	Strategie générale de test	10
6.1.	Information à tester	10
6.2.	Techniques de test	10
7.	Specification de test	11
8.	Organisation de test	13
8.1.	Fonctions	13
8.2.	Fournitures	13
8.3.	Critères généraux de succès / échec	13
9.	Environnement de test	14
9.1.	Configuration matérielle	14
9.2.	Configuration logicielle	14
9.3.	Outils de développement	14
9.4.	Outils de test	14
10.	Planification	15
10.1.	Contraintes identifiées	15
10.2.	Critères généraux de suspension et conditions générales de reprise	15
10.2.1.	Critères de suspension	15
10.2.2.	Conditions de reprise	15

1. INTRODUCTION

1.1.Avant propos.

Ce document a été développé par les étudiants de Master M1 de l'Institut Universitaire Professionnalisé (IUP) « Ingénierie des Systèmes d'Information » (ISI) de l'Université Paul Sabatier dans le cadre du bureau d'études.

1.2.Objet

Ce document présente la stratégie, l'organisation, l'environnement et la planification des activités de test de validation du logiciel Apes2.

Il identifie notamment les éléments à tester, les tâches correspondantes, les ressources requises pour ces tâches et les risques associés au présent plan.

Les cas et procédures de test sont fournis dans le cahier de validation du logiciel [CVL].

1.3.Responsabilités

Les responsables de ce document sont les membres de l'équipe de test.

2. DOCUMENTS APPLICABLES ET DE REFERENCE

2.1. Documents applicables

- [12207] **NF ISO/CEI 12207 : 1995 (F)**
Traitement de l'information – Ingénierie du logiciel – Processus du cycle de vie du logiciel
AFNOR, 1995
- [9126] **NF ISO/CEI 9126 : 1992**
Évaluation des logiciels - Caractéristiques de qualité et lignes directrices pour leur utilisation.
AFNOR, 1992
- [NF LOG] **NF LOGICIEL**
Règlement de la marque NF Logiciel - Exigences pour le dossier de test du logiciel – Annexe 1.2
AFNOR, 1996
- [14598-5] **ISO/IEC 14598-5:1998(E)**
Information technology - Software Product Evaluation - Part 5: Process for Evaluators.
ISO/IEC, 1998
- [14598-6] **ISO/IEC FDIS 14598-6:1999**
Information technology - Software Product Evaluation - Part 6: Documentation of evaluation modules.
ISO/IEC, 1999

2.2. Documents de référence

- [DCU] Document des cas d'utilisation du logiciel Apes2.
- [SS] Spécifications supplémentaires du logiciel Apes2.
- [CVL] Cahier de validation du logiciel Apes2.

3. TERMINOLOGIE

3.1. Glossaire

La terminologie se conforme aux documents applicables mentionnés au chapitre 2.

Pour rappel [NF LOG]:

Stratégie de validation : description des objectifs de test, description et justification des objectifs en termes de couverture d'essai, ainsi que des méthodes qui vont permettre d'atteindre ces objectifs

Cas de test : instructions écrites à l'usage du responsable d'essai qui spécifient la manière dont une fonction ou une combinaison de fonctions doit être testée ou dont il convient qu'elle le soit. Un cas de test contient des informations détaillées sur les points suivants :

- l'objectif du test ;
- les fonctions à tester ;
- l'environnement d'essai et autres conditions (configuration détaillée et travaux préparatoires) ;
- les données de test ;
- le mode opératoire ;
- le comportement attendu du programme.

Scénario de test : ensemble autonome et cohérent d'interactions avec le logiciel regroupant un ou plusieurs tests élémentaires, et les éventuelles phases préparatoire et finale de ces tests élémentaires.

Test élémentaire : test spécifiant les interactions avec le logiciel contribuant à vérifier un objectif de test particulier et une fonction particulière.

Procédure de test : expression d'un test élémentaire en une succession de commandes ou d'instructions directement applicables par les personnes chargées d'exécuter les tests.

3.2. Acronymes et abréviations

AQ	Assurance Qualité
BE	Bureau d'études
IUP	Institut Universitaire Professionnalisé
IHM	Interface Homme/Machine
ISI	Ingénierie des Systèmes d'Information
V&V	Vérification et Validation
IT	Itération
RUP	Rational Unified Process
CU	Cas d'utilisation
PVL	Plan de Validation Logiciel

4. EXIGENCES DE TEST

4.1.Exigences communes

Ce paragraphe rappelle les exigences communes à l'ensemble des outils développés dans le cadre du BE ISI M1.

- [EX-U-01] Les outils doivent posséder une interface (menus, messages d'erreur,...) en français.
- [EX-P-01] L'internationalisation doit pouvoir se faire sans re-compilation, uniquement en modifiant un fichier de ressources. Il est recommandé d'en proposer une version en anglais.
- [EX-U-02] Les outils doivent posséder un menu Préférences dans lequel on peut au moins choisir la langue et le chemin du répertoire où seront mis les fichiers produits.
- [EX-U-03] Les outils doivent posséder un menu Aide avec une rubrique à propos dans laquelle apparaît le nom de l'outil, le numéro de version et la date de fabrication.
- [EX-M-01] Les composants utilisés doivent être cités dans la rubrique A propos..., ainsi que les informations de licence.
- [EX-P-02] Les outils doivent pouvoir être utilisés sous Windows sans installation ni écriture dans le registre.
- [EX-P-03] Les outils doivent être fournis sous forme de fichier jar pouvant être lancé directement.
- [EX-R-01] Les outils doivent se lancer en moins de 3 secondes.
- [EX-P-04] Les outils doivent fonctionner au moins avec Java 1.4.2.

Dans le cadre du présent plan de test, sont considérées comme exigences de test :

- [EX-U-01]
- [EX-U-02]
- [EX-M-01]
- [EX-R-01]

4.2.Exigences spécifiques

[DCU] et [SS] spécifient la combinaison de recommandations, contraintes et exigences spécifiques à satisfaire par le logiciel.

Dans le cadre du présent plan de test, sont retenues comme exigences de test :

- [EX-F-01] Ouvrir un nouveau composant
- [EX-F-02] Ouvrir un composant sauvegardé
- [EX-F-03] Ajouter une définition de travail à un composant
- [EX-F-04] Ajouter une activité à une définition de travail
- [EX-F-05] Ajouter un élément dans un diagramme
- [EX-F-06] Ajouter un diagramme de responsabilités
- [EX-F-07] Ouvrir des interfaces
- [EX-F-08] Ajouter un lien entre un produit et un composant dans le diag de contexte

- [EX-F-09] Ajouter un état à un produit
- [EX-F-10] Imprimer un diagramme
- [EX-F-11] Gérer le texte et les couleurs
- [EX-F-12] Ajouter un produit avec état dans un diagramme à partir de l'arbre
- [EX-F-13] Vérifier la consistance de processus
- [EX-F-14] Déplacer un lien
- [EX-F-15] Ajouter des notes à un diagramme
- [EX-NF-01] Changer de langue
- [EX-NF-02] Copier Coller
- [EX-NF-03] Couper Coller

5. DESCRIPTION DU LOGICIEL

5.1. Domaine d'application et objectifs

APES2 est un outil d'ingénierie de processus pour les ingénieurs processus, les ingénieurs qualité et les chefs de projet qui désirent produire des composants de processus.

L'objectif est de gérer des composants réutilisables, de créer un modèle pour les composants et d'avoir un système permettant de vérifier la conformité du composant.

5.2. Information

5.2.1. Programmes

Le logiciel est composé des programmes/exécutables suivants:

[EXE1]	Exécutable Apes2.jar
--------	----------------------

Dans le cadre du présent plan, Apes2.jar est retenu.

5.2.2. Documentation Utilisateur

Les éléments suivants constituent la documentation Utilisateur du logiciel.

[MU]	Manuel Utilisateur
------	--------------------

Ces éléments ne font l'objet d'aucun test dans le cadre dans le présent plan.

5.2.3. Documentation technique

Les éléments suivants constituent la documentation technique du logiciel.

[DCU]	Document des cas d'utilisation
[DV]	Document Vision
[LR]	Liste des risques
[PCV]	Plan de cycle de vie
[PD]	Plan de développement
[SS]	Spécifications supplémentaires
[DAL]	Document d'architecture logiciel
[PI]	Plan des itérations
[EI]	Evaluation des itérations
[EXC]	Exigences complémentaires
[PVL]	Plan de validation logiciel
[CVL]	Cahier de validation logiciel

Ces éléments ne font l'objet d'aucun test dans le cadre dans le présent plan.

5.2.4. Code Source

5.2.4.1. Code spécifique

Le code source du logiciel est composé des paquetages suivants :

[INTERFACE1]	apes.ui
[APPLICATION1]	utils
[APPLICATION2]	apes
[APPLICATION3]	apes.ui.tools
[APPLICATION4]	apes.ui.actions
[APPLICATION5]	apes.processing
[APPLICATION6]	apes.adapters
[DOMAINE1]	apes.model

Ces éléments ne font l'objet d'aucun test dans le cadre le présent plan.

5.2.4.2. Code réutilisé

[LIB1]	Jgraph
[LIB2]	JSX
[LIB3]	SAX

S'agissant de code réutilisé, ces items ne font l'objet d'aucun test dans le cadre du présent plan.

5.3.Découpage en composants

Sans Objet.

6. STRATEGIE GENERALE DE TEST

Ce chapitre résume la stratégie de test mise en place pour la validation du logiciel. Il précise notamment, pour chaque caractéristique logicielle [9126], les informations logicielles concernées ainsi que les techniques de test retenues.

6.1.Information à tester

Pour chaque caractéristique logicielle, le tableau ci-dessous précise l'information logicielle à tester (X) :

Table : Informations à tester par critères :

Information	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Programmes et données	X		X	X		
Documentation Utilisateur						
Documentation Technique						
Code source						

6.2.Techniques de test

Le tableau ci-dessous résume les techniques de test sélectionnées en fonction des exigences de test et des informations à tester.

Techniques de test	Capacité fonctionnelle	Fiabilité	Facilité d'utilisation	Rendement	Maintenabilité	Portabilité
Test fonctionnel	X					
Test de partition						
Graphes Causes-Effets						
Test des valeurs limites						
Test de stress						
Test structurel						
Inspection de code						
Inspection de l'IHM			X			
Vérification de règles de programmation						
Test de performance				X		
Test de charge						
Mesure de la complexité du code						
Analyse du flot de contrôle						

Chaque technique de test retenue fait l'objet d'une spécification détaillée présenté au chapitre suivant.

7. SPECIFICATION DE TEST

Ce chapitre raffine la stratégie de test pour chaque technique de test sélectionnée.

Identificateur :	FUNC
Technique de test :	Test fonctionnel
Exigences testées :	
[EX-F-01]	Ouvrir un nouveau composant
[EX-F-02]	Ouvrir un composant sauvegardé
[EX-F-03]	Ajouter une définition de travail à un composant
[EX-F-04]	Ajouter une activité à une définition de travail
[EX-F-05]	Ajouter un élément dans un diagramme
[EX-F-06]	Ajouter un diagramme de responsabilités
[EX-F-07]	Ouvrir des interfaces
[EX-F-08]	Ajouter un lien entre un produit et un composant
[EX-F-09]	Ajouter un état à un produit
[EX-F-10]	Imprimer un diagramme
[EX-F-11]	Gérer le texte, les couleurs et les interfaces.
[EX-F-12]	Ajouter un produit avec état dans un diagramme à partir de l'arbre
[EX-F-13]	Vérifier la consistance de processus
[EX-F-14]	Déplacer un lien
[EX-F-15]	Ajouter des notes à un diagramme
Information sous test :	
[EXE1]	Exécutable Apes2.jar
Cas de test : On peut remarquer que l'on fait un cas de test spécifique par exigence.	
Niveau de complétude : 100% des exigences mentionnées en 4.2.	
Critères de succès / échec : Le test est excellent si aucune erreur n'est survenue. Si une erreur même non critique survient le test est faiblement réussi.	
Critères de suspension et conditions de reprise : Voir en 10.2	

Identificateur :	NFUNC
Technique de test :	Test non fonctionnel
Exigences testées :	
[EX-NF-01]	Changer de langue
[EX-NF-02]	Copier Coller
[EX-NF-03]	Couper Coller
Information sous test :	
[EXE1]	Exécutable Apes2.jar
Cas de test : On peut remarquer que l'on fait un cas de test spécifique par exigence.	
Niveau de complétude : 100% des exigences mentionnées en 4.2.	
Critères de succès / échec : Le test est excellent si aucune erreur n'est survenue. Si une erreur même non critique survient le test est faiblement réussi.	
Critères de suspension et conditions de reprise : Voir en 10.2	

Identificateur :	INSP
Technique de test :	Inspection de l'IHM
Exigences testées :	
[EX-U-01]	Les outils doivent posséder une interface (menus, messages d'erreur,...) en français.
[EX-U-02]	Les outils doivent posséder un menu Préférences dans lequel on peut au moins choisir la langue et le chemin du répertoire où seront mis les fichiers produits.
[EX-M-01]	Les composants utilisés doivent être cités dans la rubrique A propos..., ainsi que les informations de licence.
Information sous test :	
[EXE1]	Exécutable Apes2.jar
Cas de test : On peut remarquer que l'on fait un cas de test général pour ces exigences.	
Niveau de complétude : <i>Un cas de test pour l'ensemble des exigences retenues.</i>	
Critères de succès / échec : Les tests sont excellents si les principes ergonomiques d'IHM sont respectés : pas plus de trois niveaux de sous-menus, les fonctionnalités non disponibles sont grisées ou bien n'apparaissent pas, présence d'info-bulles, messages d'erreur explicites,... Le résultat du test varie de bon à moyen suivant le défaut d'IHM, le test est faiblement réussi si l'IHM est illisible ou inexploitable.	
Critères de suspension et conditions de reprise : Voir en 10.2	

Identificateur :	PERF
Technique de test :	Test de performance
Exigences testées :	
[EX-R-01]	Les outils doivent se lancer en moins de 3 secondes.
Information sous test :	
[EXE1]	Exécutable Apes.jar
Cas de test :	
[TP01]	Test de performance
Niveau de complétude : <i>Un cas de test par exigence retenue.</i>	
Critères de succès / échec : Le test est excellent si l'outil respecte tous les temps de réponse des exigences. Plus de 3 secondes de retard rend le test faible.	
Critères de suspension et conditions de reprise : Voir en 10.2	

8. ORGANISATION DE TEST

Ce chapitre présente l'organisation mise en place pour réaliser le présent plan de test.

8.1. Tâches

Le découpage en tâche pour réaliser le présent plan de test est le suivant :

- spécifier les tests.
- développer/écrire les tests.
- exécuter les tests.

L'équipe de test (Chef testeur : Nicolas DELMAS et Testeur1 : Isabelle CASSAGNEAU) est en charge de mener ces tâches.

8.2.Fonctions

Le tableau ci-dessous indique les différentes fonctions requises pour exécuter le présent plan de test ainsi que les ressources initialement allouées pour chacune de ces fonctions.

Table : Fonctions et ressources allouées:

Fonctions	Ressources allouées	Organisation
RC - Représentant Client	Claude AUBRY, Henri MASSIE	IUP ISI
RU - Représentant Utilisateurs	Claude AUBRY, Henri MASSIE	IUP ISI
CP - Chef de Projet	Delphine DE ROSSO	
CT – Chef testeur	Nicolas DELMAS	
T – Testeur1	Isabelle CASSAGNEAU	
ED - Expert Domaine	Lionel PETIT	

8.3.Fournitures

Nous devons fournir :

- jeux de données : exemple.apes et un fichier composant vide avec interfaces d'IEPP,
- un cahier de test rempli,
- un cahier de test vierge,
- le présent plan.

8.4.Critères généraux de succès / échec

Pour faire une synthèse globale, nous pouvons dire que la validation est prononcée si les tests fonctionnels FUNC sont bons, les tests non fonctionnels NFUNC, l'inspection de l'IHM INSP et la partie test de performance PERF sont moyens.

On ne rejettera le logiciel que s'il n'assure pas les fonctionnalités indispensables liées à la modélisation d'un composant de processus.

9. ENVIRONNEMENT DE TEST

Ce chapitre décrit l'environnement requis pour exécuter le présent plan de test. Certains tests peuvent nécessiter des besoins complémentaires. Le cas échéant, se référer à la rubrique correspondante dans les spécifications des cas de tests fournis dans [CVL].

9.1. Configuration matérielle

La configuration matérielle suivante est requise pour exécuter le présent plan de test :

- Un PC

9.2. Configuration logicielle

La configuration logicielle suivante est requise pour exécuter le présent plan de test :

- Java 2 Runtime Environnement SE v1.4.2
- Un système d'exploitation Windows ou Linux

9.3. Outils de développement

Les outils de développement suivants sont requis pour exécuter le présent plan de test :

- Sans objet.

9.4. Outils de test

Les outils de test suivants sont requis pour exécuter le présent plan de test :

- Sans objet.

10. PLANIFICATION

Ce chapitre décrit la planification requise pour exécuter le présent plan de test.

10.1. Contraintes identifiées

- Spécification des tests en début d'itération.
- Développement des tests en milieu d'itération.
- Exécution des tests de validation la semaine précédent la livraison.

10.2. Critères généraux de suspension et conditions générales de reprise

10.2.1. Critères de suspension

Les critères généraux de suspension du présent plan de test sont les suivants :

- Une erreur bloquante.
- Chaque test est en suspension si les tests pré-requis de ce dernier ne peuvent s'exécuter entièrement.

10.2.2. Conditions de reprise

La condition générale de reprise du présent plan de test est la suivante :

- Livraison d'un nouvel exécutable.