

APES (APES is a Process Engineering Software)

Spécifications supplémentaires

Version 2.0

Auteur : Isabelle Cassagneau

Table des Révisions

Révision	Date	Auteur(s)	Description
2.0	18/11/2003	Isabelle Cassagneau	Modification suite à la revue de l'IT0
1.0	11/11/2003	Isabelle Cassagneau	Création du document

Table des Matières

Chapitre 1 : Introduction.....	1
1. Objectif.....	1
2. Portée.....	1
3. Références.....	1
Chapitre 2 : Utilisabilité.....	2
1. Installation.....	2
2. Convivialité.....	2
Chapitre 3 : Fiabilité.....	3
1. Pourcentage de défauts.....	3
Chapitre 4 : Performance.....	4
1. Temps de réponse.....	4
2. Capacité.....	4
3. Mode de dégradation.....	4
4. Utilisation des ressources.....	4
Chapitre 5 : Supportabilité.....	5
1. Maintenabilité.....	5
2. Portabilité.....	5
3. Evolutivité.....	5
Chapitre 6 : Contraintes de conception.....	6
1. Langage de programmation.....	6
2. Bibliothèques utilisées.....	6
3. Format d'échange.....	6
Chapitre 7 : Documentation utilisateur et exigences de système d'aide.....	7
Chapitre 8 : Exigence de licence, loi, copyright.....	8
1. Licences.....	8

Chapitre 1

Introduction

1. Objectif

Les spécifications supplémentaires recueillent les exigences qui ne peuvent pas être associées au modèle des cas d'utilisation. Parmi ces exigences :

- les besoins non fonctionnels : utilisabilité, fiabilité, performance et supportabilité
- les contraintes liées à l'environnement, au système d'exploitation, à la compatibilité, au choix de conception
- les standards applicables

2. Portée

Les rôles ou ensembles de rôles suivants utilisent les spécifications supplémentaires :

- les analystes créent et maintiennent à jour les spécifications supplémentaires, qui servent de moyen de communication entre analystes, clients, et développeurs.
- les développeurs utilisent les spécifications supplémentaires en tant que référence lorsqu'ils définissent les responsabilités, les opérations, et attributs des classes, et lorsqu'ils ajustent les classes pour l'environnement d'implémentation.
- le chef de projet se sert des spécifications supplémentaires comme une entrée de la planification des itérations.
- les testeurs utilisent les spécifications supplémentaires pour vérifier la conformité du système.

3. Références

- Document de vision
- Glossaire

Chapitre 2

Utilisabilité

1. Installation

La procédure d'installation et de paramétrage du logiciel doit être automatique, sans écriture dans le registre. Le logiciel doit être fourni sous forme de fichier jar et sa mise en place doit pouvoir être effectuée directement, sans l'aide de l'équipe de développement.

2. Convivialité

L'application s'adresse à des concepteurs connaissant la gestion de processus. Le manuel d'utilisation ne doit pas être nécessaire pour une première prise en main du logiciel. Les interfaces vont respecter les standards des interfaces graphiques Java.

Chapitre 3

Fiabilité

1. Pourcentage de défauts

L'application devra être testée au maximum afin d'éviter les bugs critiques comme la perte complète de données, impossibilité totale d'utiliser l'outil, impossibilité totale d'utiliser certaines fonctionnalités du système.

Chapitre 4 Performance

1. Temps de réponse

L'utilisateur sera immédiatement averti qu'un traitement est en cours pour toute action qui nécessite ou risque de nécessiter un temps important (ex. : Vérification de la consistance d'un processus). La durée maximale d'attente supportable par un utilisateur dans le cas d'un long traitement est fixée à 3 secondes. Le langage Java étant gourmand et exigeant en ressources, on n'imposera pas de contraintes au niveau des flux.

2. Capacité

L'application étant personnelle, seul un utilisateur à la fois l'utilisera. Il n'y aura donc pas de problème de capacité.

3. Mode de dégradation

Etant donné qu'à une version du logiciel ne correspond qu'une personne, l'utilisateur est responsable de ses actions. Toute dégradation du logiciel par l'utilisateur ne sera donc pas pris en compte.

4. Utilisation des ressources

L'utilisateur devra réserver un espace disque d'environ 15Mo pour l'application et le moteur Java. Il devra également tenir compte des sauvegardes qu'il voudra effectuer.

Chapitre 5

Supportabilité

1. Maintenabilité

Les années à venir, les bureaux d'études pourront avoir pour but de compléter le logiciel. Le code source sera mis à disposition des futurs développeurs. L'utilisateur pourra avoir accès à une version de mise à jour après la maintenance des bureaux d'études à venir.

2. Portabilité

Le logiciel pourra être utilisé au moins sur Unix et Windows.

3. Evolutivité

Il faudra tenir compte du fait que le modèle SPEM, fourni pour la vérification de la conformité du processus, pourra évoluer. Il faudra donc prévoir une implémentation de ce modèle pouvant être adaptée aisément.

Chapitre 6

Contraintes de conception

1. Langage de programmation

L'application devra obligatoirement être codée en Java ce qui obligera l'équipe de développeurs d'être doté d'un moteur Java.

2. Bibliothèques utilisées

Il devra être exécutable à partir des bibliothèques d'exécution Java 2 version 1.4 et plus. Java étant supporté sur la plupart des systèmes d'exploitation. Il suffira au système d'avoir les bibliothèques Java « RunTime » installées.

3. Format d'échange

Utilisation du XML format exportable pour la persistance des données.

Chapitre 7

Documentation utilisateur et exigences de système d'aide

La distribution de l'application sera accompagnée d'un manuel d'utilisation.

Chapitre 8

Exigence de licence, loi, copyright

1. Licences

Tous les éléments logiciels doivent être distribués sous licence GNU GPL (GNU General Public License) version 2. Toute la documentation doit être livrée sous la licence GNU FDL (GNU Free Documentation License) version 1.1.

Des informations sont disponibles sur le site de la Free Software Foundation (FSF) :

- <http://www.fsf.org/copyleft/gpl.html> (licence GNU GPL)
- <http://www.fsf.org/copyleft/fdl.html> (licence GNU FDL)

Les informations de licence doivent être citées dans la rubrique à propos.