

A large, faint watermark of the ULB seal is visible on the left side of the slide. The seal is circular and contains a central sunburst, two crossed torches, and the Latin motto "VIA VINCERE TENEBRAS".

# Software Requirements Document Do's and Don't's

Anthony Cnudde  
Abel Laval

INFO-F-204 & INFO-F-209  
[anthony.cnudde@ulb.be](mailto:anthony.cnudde@ulb.be)  
[abel.laval@ulb.be](mailto:abel.laval@ulb.be)

5 novembre 2021

A decorative horizontal bar at the bottom of the slide, composed of several colored squares in black, grey, magenta, green, blue, red, orange, and purple.

# Qu'est-ce que c'est ?

## Définition - IEEE 930-1998

Le SRD (aussi appelé SEL) est un document qui sert à spécifier les exigences d'un logiciel, d'un programme ou d'un progiciel en particulier, qui exécute certaines fonctions dans un environnement précis.

## À quelles questions doit-il répondre ?

- ▶ Que doit faire le logiciel ? ⇒ **Fonctionnalités**
- ▶ Quelles types de liens doit-il y avoir entre le logiciel et les utilisateurs, le matériel du système, les autres matériels et les autres logiciels ? ⇒ **Interfaces externes**
- ▶ Quelle doit être la vitesse, le degré de disponibilité, le délai de réponse et le délai de récupération des diverses fonctions logicielles ? ⇒ **Performances**
- ▶ Y a-t-il des contraintes dont il faut tenir compte (normes, langages d'implémentation) ? ⇒ **Contraintes**

# Structure de base du SRD pour le PA2

1. Introduction
  - 1.1 But
  - 1.2 Glossaire
  - 1.3 Historique
2. Besoins utilisateur
  - 2.1 Fonctionnels
  - 2.2 Non fonctionnels
3. Besoins système fonctionnels
  - 3.1 Fonctionnels
  - 3.2 Non fonctionnels
4. Design et fonctionnement du système
5. Annexes

# Introduction

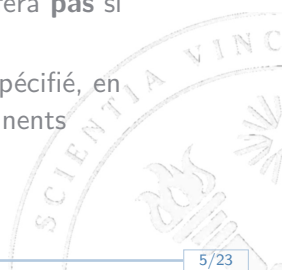


# But du projet

## But

Décrire et délimiter le but du SRD, et identifiez par leur(s) nom(s) le(s) logiciel(s) à produire

- ▶ Spécifiez l'audience visée  $\Rightarrow$  Qui va lire ce document ?
- ▶ Expliquez ce que chaque logiciel fera (et ne fera **pas** si nécessaire)
- ▶ Décrivez les applications de chaque logiciel spécifié, en incluant les avantages, objectifs et buts pertinents



## But

Spécifier tous les termes utilisés dans le reste du document pour pouvoir comprendre de façon correcte le SRD.

**N.B. :** Cette information peut être contenue dans une annexe et vous pouvez référencer à l'annexe en question.

## Exemples :

- **Student :** A severely sleep deprived person attending a university. Said deprivation normally due to studying, but more often, social life. Students are also commonly associated with procrastination on school assignments which then results in further lack of sleep .

# Historique

**But**

Garder une trace des différentes versions du SRD, de ses auteurs et des modifications

Numéro de version	Nom	Modifications	Date
0.1		Besoin utilisateur	3/12/16
0.2		Besoin système	4/12/16
0.3		Gestion pj + gestion argent	4/12/16
0.4		Gestion fin de partie et score	8/12/16
0.5		Besoin utilisateur: Non fonctionnels	11/12/16
0.6		But du projet + détail besoin utilisateur	11/12/16
0.7		Ajout use case diagram : utilisateur + système (connexion, matchmaking)	17/12/16
0.8		Ajout use case diagram : besoin système	17/12/16
0.9		Ajout du glossaire	18/12/16
0.10		Gestion de la difficulté + Commentaires + Revue des tours	18/12/16
0.11		Inclusion diagrammes séquences	18/12/16
0.12		commentaires diagrammes séquences	18/12/16
0.13		ajouts commentaires diagrammes séquences	18/12/16
0.14		design système	18/12/16
0.15		activity diagram matchmaking	19/12/16
0.16		besoin système: non-fonctionnels	19/12/16

Date	Version	Description	Auteur
27/11/2015	0.0	<ul style="list-style-type: none"> <li>• Use case "Gameplay"</li> <li>• Use case "Interface"</li> <li>• Diagramme de Classes</li> <li>• Diagramme de séquences.</li> </ul>	Totaleté du groupe :
09/12/2015	1.0	Rectification du travail	Totaleté du groupe

(a) Un bon exemple

(b) Sans commentaires...

FIGURE – Exemples d'historique

# Besoins





# Besoins fonctionnels

## But

Definition des actions principales que le logiciel doit exécuter, que ce soit au point de vue de l'utilisateur ou du système.

- ▶ Répond à : **Qu'est-ce que l'utilisateur peut faire avec le logiciel ?**
- ▶ Les exigences fonctionnelles sont exprimées sous la forme suivante : « Le système doit... », « L'utilisateur peut... »
- ▶ Il peut être utile de diviser ces exigences en sous-fonctions ou en sous-processus. Cela n'implique pas nécessairement que le logiciel sera conçu selon ces mêmes divisions.

# Non fonctionnels

## But

Définition des besoins optionnels (performance, vitesse, etc.) ou liés à des contraintes de matériel, de caractéristiques d'utilisateur,...

- ▶ Répond à : **Comment le logiciel doit réaliser les fonctionnalités ?**
- ▶ Exemples :
  - ▶ Exploitation en parallèle
  - ▶ Exigences de fiabilité
  - ▶ Considérations relatives à la sécurité et à la sûreté (GDPR)
  - ▶ Adaptation graphique pour daltonien
- ▶ Limites imposées par le matériel (p. ex. : temps de réponse, dispositifs d'input)

# Besoins Utilisateur



## Contenu :

Toutes les interactions possibles de l'utilisateur avec le logiciel.

**Type de diagrammes conseillé :** Use Case

## Exemples :

- ▶ Connexion
- ▶ Démarrer une partie
- ▶ Actions durant une partie
- ▶ Visualisation classement



# Non-fonctionnels

## Contenu :

Besoins éventuels liés à l'utilisateur du jeu

## Exemples :

- ▶ Contraintes liées à l'accès au jeu
- ▶ Contraintes liées à l'apparence de l'interface



# Besoins Système



# Fonctionnels

## Contenu :

Toutes les fonctionnalités résultantes de l'interaction de l'utilisateur avec le logiciel.

**Type de diagrammes conseillé :** Use Case

## Exemples :

- ▶ Connexion
- ▶ Menu principal
- ▶ Gestion des comptes
- ▶ Création d'une partie
- ▶ Gestion d'une partie
- ▶ Mise à jour du classement



# Non fonctionnels

## Contenu :

Contraintes liées au matériel et/ou à des caractéristiques optionnelles.

## Exemples :

- ▶ Disponibilité
- ▶ Sécurité
- ▶ Maintenabilité
- ▶ Transférabilité
- ▶ Scalabilité
- ▶ Système d'exploitation







# Design et fonctionnement du système



## But

Décrit et conceptualise le logiciel de façon statique.

- ▶ **Type de diagrammes conseillé** : Diagramme de classes
- ▶ Vous pouvez diviser le système en plusieurs diagrammes de classes ou travailler avec un code couleur pour délimiter les différentes divisions du système.



# Fonctionnement

## But

Décrit le fonctionnement du système de façon dynamique

- ▶ **Types de diagrammes conseillés** : Diagramme de séquences, de collaboration et d'activité
- ▶ Chaque fonctionnalité principale du système doit être décrite pour comprendre les différents messages, entrants et sortants communiqués entre les différentes parties du système.



# Pour résumer...

## Modélisation statique :

- ▶ Diagramme Use case  $\Rightarrow$  Fonctionnalités
- ▶ Diagramme de classes  $\Rightarrow$  Entités et leurs relations

## Modélisation dynamique :

- ▶ Diagramme de collaboration/séquence  $\Rightarrow$  Déroulement séquentiel des opérations
- ▶ Diagramme d'activité  $\Rightarrow$  Flux des données à travers de l'application



# Annexes



## But

Contenir des informations supplémentaire qui pourraient aider à la compréhension des exigences mais qui ne sont pas nécessaires à leur définition

## Exemples :

- ▶ Des spécimens de formats d'entrée/sortie
- ▶ Des renseignements d'appui et de l'information documentaire qui peuvent aider les lecteurs à comprendre la SEL.



# Do's and Don't's

## Do :

- ▶ Mettre à jour **régulièrement** le SRD
- ▶ Respecter les notations standard UML
- ▶ Collaborer sur le document
- ▶ Demander des clarifications en cas de incompréhension

## Don't :

- ▶ Insérer des diagrammes incomplets/redondants
- ▶ Écrire des descriptions vagues

