**GABARIT DU RAPPORT DE SPRINT**

# Page titre

Indiquant :

1. Nom/titre du projet/logiciel
2. RAPPORT DU SPRINT *n*
3. Membres de l’équipe (excluant les codes permanents)

# Table des matières

# 1. Introduction

Introduction au rapport

## 1.1 Objectifs

Il s’agit des objectifs du document (ce qu’il décrit, son public-cible).

## 1.2 Vue d’ensemble du produit (reprenez le contenu de 1.2 du ConOps)

* Identifier le système logiciel par nom
* Expliquer brièvement ce qu'il doit faire et ne pas faire (limites/bordures)
* Décrire le contexte de l'application dans lequel le produit s'intègre, incluant les objectifs, les bénéfices et les retombées du projet.

## 1.3 Définitions, acronymes et abréviations

## 1.4 Documents de références (reprendre le contenu du ConOps)

Lister tous les documents, normes, rapports, etc.

## 1.5 Aperçu du document

Sommaire de ce qu’il contient et explication sur son organisation.

# 2. Description générale du logiciel (1.5 à 3 pages, max)

## 2.1 Vue d'ensemble des fonctions du produit

Ici, vous reprenez le contenu de la section 5.3 du document de présentation du concept.

Par rapport au document de présentation de concept, on va soit retrouver les mêmes (si on a visé juste la première fois), soit en retrouver plus, soit en retrouver moins, soit les présenter différemment. Tout cela est normal, puisque, rendu à la fin de l’analyse, on en sait plus.

Les fonctionnalités doivent être regroupées de façon hiérarchique. Par exemple, pour nos amis des assurances :

* Functionnalités assuré(e)s :
  + Soumettre une réclamation: <description sommaire>
  + Suivre l’état d’une réclamation: <description sommaire>
  + Contester le résultat d’une réclamation: <description sommaire>
  + ...
* Fonctions agents/ajusteurs de réclamation :
  + Prise en charge d’une réclamation: <description sommaire>
  + Traitement d’une réclamation: <description sommaire>
  + Traitement d’une demande de révision: <description sommaire>
  + ...

## 2.2 Contenu des sprints

Une description sommaire du contenu des sprints planifié initialement (ce que vous avez montré lors de la présentation du ConOps). Le product backlog (fourni en annexe) montre les informations détaillées, mise à jour au fur et à mesure.

# 3. contenu du sprint actuel (6 à 10 pages. max)

## 3.1 Cas d’utilisation implantés dans le sprint

Listez les cas d’utilisation, et choisissez en deux que vous documenterez de façon détaillée.

Pour chaque cas d'utilisation, il faudra décrire:

1. le cas d'utilisation, sous forme textuelle, selon le format 'long' (càd, les étapes du cas d'utilisation sont proprement numérotées, et les scénarios alternatifs ou extensions sont clairement documentés).
2. un diagramme de séquence système. Il s’agit ici des diagrammes de séquence *système*. C'est-à-dire que vous aurez une barre pour chaque acteur, et **une seule barre pour le système** perçu comme une boite noire. C’est une transcription visuelle assez directe du scénario du cas d’utilisation, à cette différence près : les messages envoyés par l’acteur au système sont représentés de façon précise (nom, paramètres), et correspondront aux contrats (au sens de Larman) que vous aurez à documenter.

Aussi, il faudra lister l’ensemble des fonctions système utilisées par les deux cas d'utilisation.

Finalement, il faudra documenter *deux* des fonctions systèmes identifiées, de façon détaillée, selon le format de *contrat* décrit ci-bas:

|  |  |
| --- | --- |
| **Signature** | <type retour> <nom fonction>(<nom param\_1>,…,<nom param\_i>) |
| **Description** | Ce que fait l’opération |
| **Références** | Où (quel cas d’utilisation/étape) l’opération est elle utilisée |
| **Entrées** | Une énumération des paramètres à l’entrée. Pour chaque paramètre, on décrira :   * Le nom du paramètre * Un type abstrait (sans rentrer dans les détails, qui vont dépendre de choix de conception) * La signification (surtout !!) du paramètre : que représente t-il |
| **Sorties** | En général une, mais des fois, il peut y en avoir plusieurs (e.g., une donnée et un « flag »), auquel cas, utilisez le même format que pour les paramètres d’entrée |
| **Exceptions** | Cas de situations exceptionnelles qu’on ne pouvait filtrer avec les pré-conditions |
| **Pré-conditions** | Hypothèses sur les valeurs des paramètres à l’entrée, et le cas échéant, sur l’état du système avant l’appel, pour que la fonction puisse être appelée et pour qu’elle remplisse son contrat |
| **Post-conditions** | Dans quel état la fonction promet-elle de laisser le système. Les post-conditions peuvent être exprimées sous la forme de :   * Création ou destruction d’objets * Modification d’attributs (d’objets ou d’associations) * Création ou destruction d’associations |
| **Notes** | Quelques notes sur la fonction, sur l’algorithme utilisé, sur la théorie sous jacente, sur les règles d’affaires à respecter, etc. |

Dans 98% des cas, il n’y aura pas grand-chose à dire dans les exceptions, puisque les pré-conditions sont supposées filtrer les entrées problématiques. Aussi, notez qu’un contrat doit avoir **soit** une sortie, soit une post-condition (effet de bord), ou les deux. En effet, **une fonction qui n'a ni sortie ni post-condition/effet de bord, ferait quoi au juste???** Si la fonction met en oeuvre un algorithme non-trivial, ça doit aller dans le champ « **Notes**».

## 3.2 Modèle de classe

Inclure ici le diagramme des classes que vous avez eues à implanter pour les cas d’utilisation de ce sprint.

En plus du diagramme de classe, il faudra joindre un mini « dictionnaire de données » où vous fournirez une description textuelle des classes, leurs attributs, et des associations, et leurs attributs. Vous utiliserez les éléments suivants :

Les éléments pour décrire une classe :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom** |  | | | |
| **Description** |  | | | |
| **Attributs** |  | | | |
|  | **Nom** | <nom attribut 1> | **Type** | <type attribut 1> |
|  | **Description** | <description attribut 1> | | |
|  | **Nom** | <nom attribut 2> | **Type** | <type attribut 2> |
|  | **Description** | <description attribut 2> | | |

Notez que vous n’êtes pas obligé d’utiliser ce même format « graphique », du moment que tous les éléments soient présents (i.e., une classe est décrite par son nom, une description textuelle, et ses attributs, et un attribut est décrit par son nom, son type, et une description textuelle).

Pour les associations :

1) avec attributs :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nom** | <nom de l’association> | | | | | |
| **Description** | <que représente t-elle?> | | | | | |
| **Entité source** | <classe source> | | **Entité destination** | | | <classe destination> |
| **Attributs** |  | | | | | |
|  | **Nom** | <nom attribut 1> | | **Type** | <type attribut 1> | |
|  | **Description** | <description attribut 1> | | | | |
|  | **Nom** | <nom attribut 2> | | **Type** | <type attribut 2> | |
|  | **Description** | <description attribut 2> | | | | |

2) sans attributs :

|  |  |  |  |
| --- | --- | --- | --- |
| **Nom** | <nom de l’association> | | |
| **Description** | < que représente t-elle?> | | |
| **Entité source** | <classe source> | **Entité destination** | <classe destination> |

# 4. PROCHAIN SPRINT (1 à 2 pages, max)

## 4.1 Revue du sprint

Revue technique (contenu du *sprint review*)

* Identifier ce qui a été complété et ce qui ne l’a pas été
* Identifier ce qui a bien fonctionné et ce qui n’a pas bien fonctionné
* Identifier les obstacles rencontrés et les solutions adoptées

Revue gestion (contenu du *sprint retrospective*)

* Évaluer les personnes, relations, processus et outils
* Identifier et classer les choses qui se sont bien déroulées et identifier des améliorations potentielles
* Créer un plan pour la mise en œuvre/adoption de ces améliorations

## 4.2 Plan pour prochain sprint

La liste des cas d’utilisation prévus pour le deuxième sprint

# 5. ANNEXES

Inclure :

1. le product backlog en date d’aujourd’hui (fin du sprint courant et début du prochain sprint)
2. Le sprint backlog du sprint courant qui se termine
3. Le sprint backlog pour le prochain sprint

### Barême

|  |  |  |
| --- | --- | --- |
| Section | Pondération | Commentaire |
| 1 | 2.5% | Reprendre le contenu du ConOps. 1 à 1.5 pages, max |
| 2  2.1  2.2 | 5%  2.5%  2.5% | Reprendre contenu (amélioré) de 5.3 du ConOps  (Résumé du) contenu textuel du product backlog |
| 3  CU  DS  CO  MO | 25%  10%  2.5%  5%  7.5% | C’est l'une des deux parties majeures du document (description comportementale/fonctionnelle) |
| 4  4.1  4.2 | 7.5%  5%  2.5% | Prendre le 4.1 au sérieux  Le contenu du sprint suivant peut être différent de ce qui était prévu initialement |