

User Stories et Diagrammes de cas d'utilisation UML

Des outils pour le recueil des exigences fonctionnelles

Master informatique parcours génie logiciel

Université de Montpellier

Plan

- 1 Introduction
- 2 Diagrammes UML
- 3 User stories
- 4 Synthèse

Ingénierie des exigences (requirements)

Phase amont de la conception des systèmes

Défi

« Réduire la fréquence et l'amplitude des problèmes de développement de systèmes en appliquant en amont des méthodes, techniques et outils qui garantissent (par leur dimension systématique) une meilleure prise en compte des exigences dans le développement des systèmes ^a ».

a. C. Salinesi, Univ. Paris Panthéon – Sorbonne, HDR, 2010

Ingénierie des exigences (requirements)

Développement motivé dans les années 90 par un constat réalisé par le Standish Group (<https://www.standishgroup.com/>) sur l'échec de nombreux projets informatiques

Principaux facteurs de succès (Chaos Report années 90)

- User involvement
- Executive management support
- **Clear Statement of Requirements**
- Proper planning
- Realistic expectations

Les facteurs de succès ont beaucoup évolué depuis (le Standish Group publie un rapport tous les 2 ans).

Ingénierie des exigences (requirements)

Standish Group Chaos Report (2020) Beyond Infinity

« Incomplete requirements cause challenged and failed projects »
serait un mythe

Que doit-on en penser ?

Depuis 10 ans, les rapports montrent la supériorité des projets développés :

- avec des méthodes **AGILE** ou proches comme SCRUM
comparativement aux projets développés de manière traditionnelle
- dans des équipes disposant d'un environnement **humain** propice
(« good sponsor, good team and good place »)

Développer l'analyse des exigences

Doit se faire de manière AGILE et continue (petites étapes, cycles courts, relation proche avec l'utilisateur)

<https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/>

<https://hennypoortman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/>



Ingénierie des exigences (requirements)

Éléments entrant en jeu

- Déterminer ce que veut un client, répondre à un appel d'offre, rédiger un contrat
- Exigences fonctionnelles (services rendus) et non-fonctionnelles/extra-fonctionnelles (ex. performance, outils imposés, normes à respecter)
- De nombreux partenaires : client, utilisateurs, chefs de projet, experts, analystes, architectes, développeurs, testeurs, etc.
- Aspects : légaux, contractuels, économique, social

Quelques langages et méthodes

- *use cases* introduits en 1987 par I. Jacobson
- scripts et scénarios (J. Hooper)
- langages formels, comme les réseaux de Petri
- langages de modélisation de buts comme KAOS (A. Dardenne) ou I* (J. Mylopoulos)
- diagrammes de cas d'utilisation en UML (1995) et méthodologie décrite dans le Unified Process (1997)
- templates de description de use case par A. Cockburn (2001)
- récits utilisateurs (user stories), cartes de récits (K. Beck 1999, M. Cohn 2004)
- adaptation des use cases au contexte AGILE, SCRUM (Use Case 2.0, I. Jacobson et al. 2011)
- cartographie des récits (J. Patton, 2014)

Plan

- 1 Introduction
- 2 Diagrammes UML**
 - Diagrammes de cas d'utilisation
- 3 User stories
- 4 Synthèse

Cas d'utilisation en UML

En UML, deux types de diagrammes principaux sont utilisés pour représenter les cas d'utilisation.

Diagramme de cas d'utilisation

- il donne une vue logique
- il organise les cas d'utilisation et les connecte aux acteurs

Diagramme de séquence

- il donne une vue chronologique
- il présente l'ordre d'appel de fonctionnalités dans un scénario (notamment le scénario nominal)

Ces diagrammes ne dispensent pas de rédiger des textes, mais ils peuvent donner une vue plus synthétique

Diagramme de cas d'utilisation

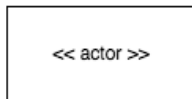
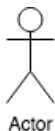
Objectif

- délimiter le système
- indiquer les acteurs
- indiquer les fonctionnalités externes
- préciser la logique d'inclusion

Diagramme de cas d'utilisation

Notation des 2 principaux concepts

- Acteur : entité extérieure au système ; acteur humain ou autre système



- Cas d'utilisation : toute manière d'utiliser le système

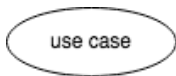
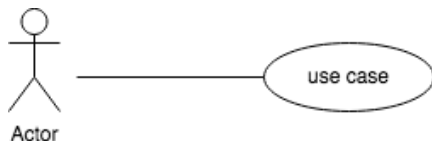


Diagramme de cas d'utilisation

Relations

- Participation (association) : un acteur participe à un cas



- Inclusion : A inclut obligatoirement B

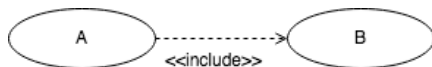


Diagramme de cas d'utilisation

Relations

- Extension : A inclut optionnellement B (B étend A ; les points d'extension peuvent être précisés)

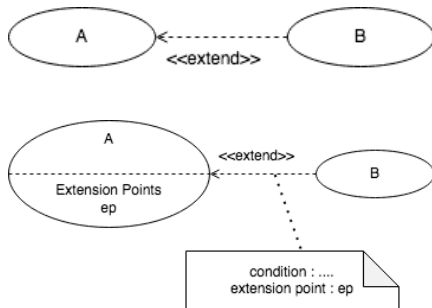


Diagramme de cas d'utilisation

Relations

- Spécialisation entre acteurs et entre cas

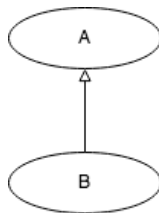


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

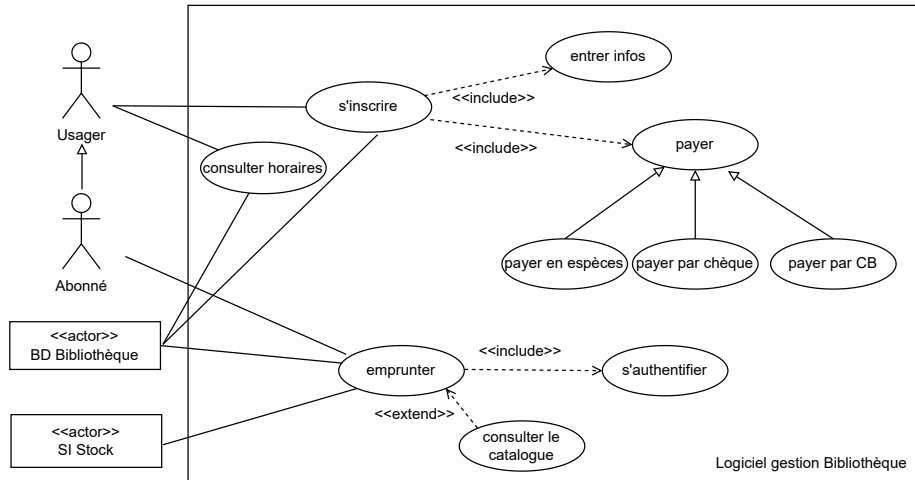


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

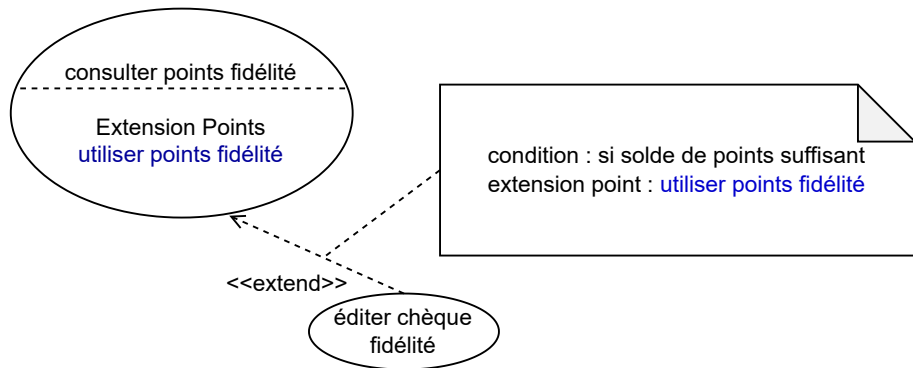


Diagramme de cas d'utilisation

Logiciel de gestion de bibliothèque

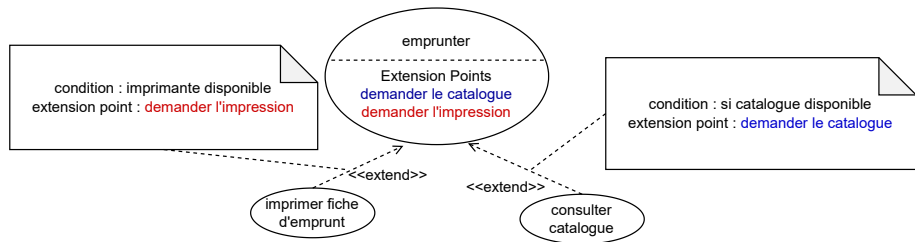


Diagramme de cas d'utilisation

Différencier extend et spécialisation entre cas : une question de point de vue et de description

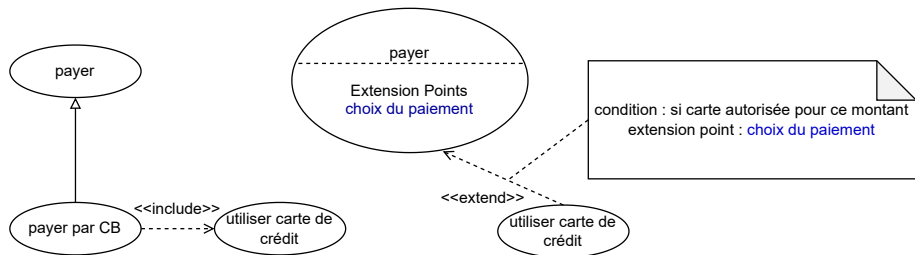
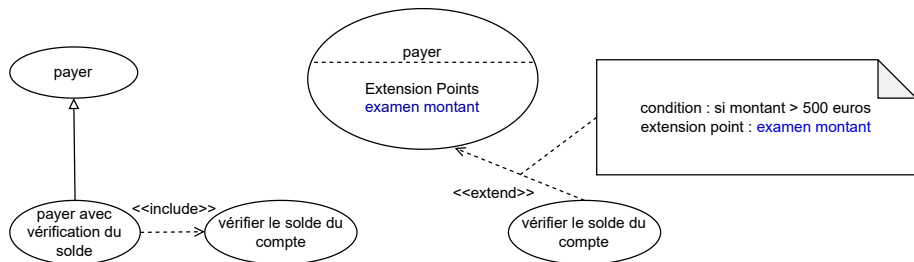


Diagramme de cas d'utilisation

Différencier extend et spécialisation entre cas : une question de point de vue et de description



Plan

- 1 Introduction
- 2 Diagrammes UML
- 3 User stories**
- 4 Synthèse

Récits utilisateurs

Principes

- Description simple d'une exigence
 - Un titre
 - Un niveau de priorité
 - Une estimation de sa complexité
 - Format "En tant que" (qui), "je souhaite" (quoi), "afin de" (pourquoi)
 - Critères INVEST
 - (qui) peut être lié à un personnage fictif représentant un utilisateur
 - Critères d'acceptation

Récits utilisateurs

Emprunt abonné ; priorité 1 ; moyen

- En tant que : abonné
- Je souhaite : emprunter un ouvrage
- Afin de : lire cet ouvrage

Vérif employé ; priorité 1 ; simple

- En tant que : bibliothécaire
- Je souhaite : vérifier le nombre d'emprunts d'un abonné
- Afin de : savoir s'il peut emprunter un exemplaire

Vérif stock ; priorité 1 ; simple

- En tant que : bibliothécaire
- Je souhaite : vérifier la présence d'un exemplaire
- Afin de : savoir si un ouvrage est disponible pour un prêt

Critères INVEST

Critères INVEST

- Indépendante
- Négociable
- Valorisable (avec de la valeur pour l'utilisateur)
- Estimable (pour la planification)
- Size (de petite taille)
- Testable

Critères d'acceptation

Critères d'acceptation menant aux tests

- Format "Etant donné que," "Lorsque," "Alors,"
- Exemple avec Vérif stock
 - En tant que : bibliothécaire
 - Je souhaite : vérifier la présence d'un exemplaire
 - Afin de : savoir si un ouvrage est disponible pour un prêt
- "Etant donné que," je suis sur la page de consultation du stock
"Lorsque," j'entre un numéro ISBN "Alors," je reçois le nombre d'exemplaires de cette référence qui sont dans le stock

Plan

- 1 Introduction
- 2 Diagrammes UML
- 3 User stories
- 4 Synthèse**

Synthèse

- Récits utilisateurs

- + Simples à décrire, à lire et à mettre en œuvre (une itération de développement AGILE), aident à discuter oralement avec les utilisateurs, peut être écrit par l'utilisateur, sert à estimer le travail
- Manque de précision, de structure et de contexte (pas de vue globale), difficile de se rendre compte de la complétude, manque de temporalité

- Cas d'utilisation et diagrammes

- + Niveau de structure et de détails importants et précis, peuvent s'adapter au contexte AGILE à condition de les décomposer et de les faire évoluer, vision globale, spécification semi-formelle
- Plus complexes à écrire (réalisés par un analyste), à lire et à modifier en cas de changement

- Les deux approches peuvent être utilisées en synergie : réaliser des cas d'utilisation et les découper en récits ou l'inverse (voir [https ://martinfowler.com/bliki/UseCasesAndStories.html](https://martinfowler.com/bliki/UseCasesAndStories.html))