

# LOG210 SÉANCE #01

## ANALYSE ET CONCEPTION DE LOGICIELS



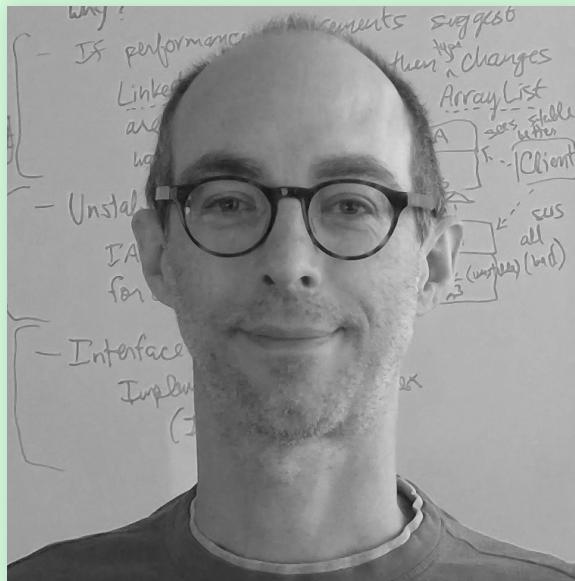
1. Présentation de l'équipe
2. Présentation personnelle
3. Présentations des étudiants
4. Faire un logiciel pour un domaine que vous ne connaissez pas
5. Outils de travail
6. Introduction à l'analyse et conception de logiciels



# ÉQUIPE



- Responsable du cours
  - Christopher Fuhrman
  - [christopher.fuhrman@etsmtl.ca](mailto:christopher.fuhrman@etsmtl.ca)



- Chargé de cours
  - Yvan Ross, M.ing
  - [cc-yvan.ross@etsmtl.ca](mailto:cc-yvan.ross@etsmtl.ca)



Utiliser Discord pour communiquer avec moi.

# COMMUNIQUER AVEC MOI



- Par discord: **objet précis**, ex. **GR01: question sur le modèle du domaine**
- Rencontres organisées de façon proactive en **dehors** de ces heures. Proposez moi au moins deux périodes de disponibilité.
  - Lun 8h30-12:00
  - Lun 13:30 - 17:00
  - Mar 18:00 - 21:30
  - Jeu 08:30 - 12:00
  - Ven 13:30 - 16:30

# CHARGÉS DE LABORATOIRE



- 01 Jeu 08:30 - 11:30 Labo A-3322
  - [hind.errahmouni.1@ens.etsmtl.ca](mailto:hind.errahmouni.1@ens.etsmtl.ca)
- 02 Lun 13:30 - 16:30 Labo A-3322
  - [edouard.laforge.1@ens.etsmtl.ca](mailto:edouard.laforge.1@ens.etsmtl.ca)
- 03 Mer 18:00 - 21:00 Labo A-3322
  - [islem.saidani.1@ens.etsmtl.ca](mailto:islem.saidani.1@ens.etsmtl.ca)

Utiliser Discord pour communiquer avec eux à propos du laboratoire.

# LOG210 SÉANCE #01

## ANALYSE ET CONCEPTION DE LOGICIELS



1. Présentation de l'équipe
2. Présentation personnelle ← s20203
3. Présentations des étudiants
4. Faire un logiciel pour un domaine que vous ne connaissez pas
5. Outils de travail
6. Introduction à l'analyse et conception de logiciels

# CHRISTOPHER



# FUHRMAN

# PRÉSENTATION PERSONNELLE

- Professeur depuis 2001 (Conception de logiciels)
- Expérience industrielle
  - 2019 : Développement Pharo.org Inria Lille
  - 2014-20 : PlantUML Gizmo
  - 2008 : Développement convertisseur XML FBI-WVU
  - 2000 : Directeur du programme NASA « ASSET »
  - 1998-1999 : Développeur chez ProLogic
  - 1989-1991 : Développeur chez Apple



# PRÉSENTATION PERSONNELLE

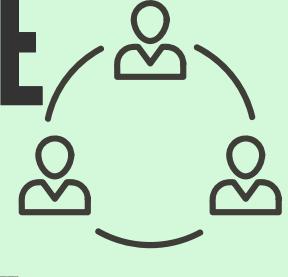


- Diplômes universitaires
  - 1988 Bachelor of Science (informatique)  
WVU
  - 1996 Docteur ès sciences (informatique technique) École Polytechnique Fédérale de Lausanne

# YVAN ROSS



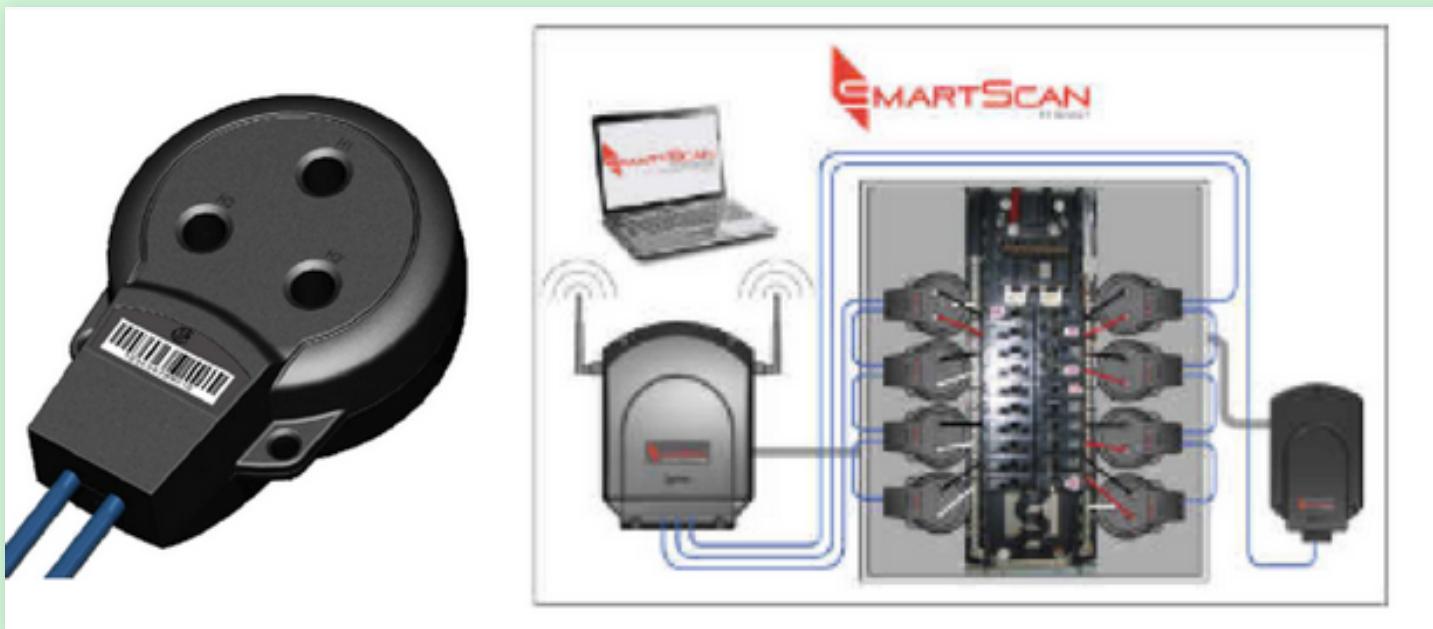
# PRÉSENTATION PERSONNELLE



- Expérience industrielle
  - 2013-: Chargé de cours ETS
  - 2010-2015: Consultant
  - 2011-2013: Web analytique cloud (ROR), gestion de projet
  - 2009-2011: Reconnaissance vocale. C++, java, ROR, Perl

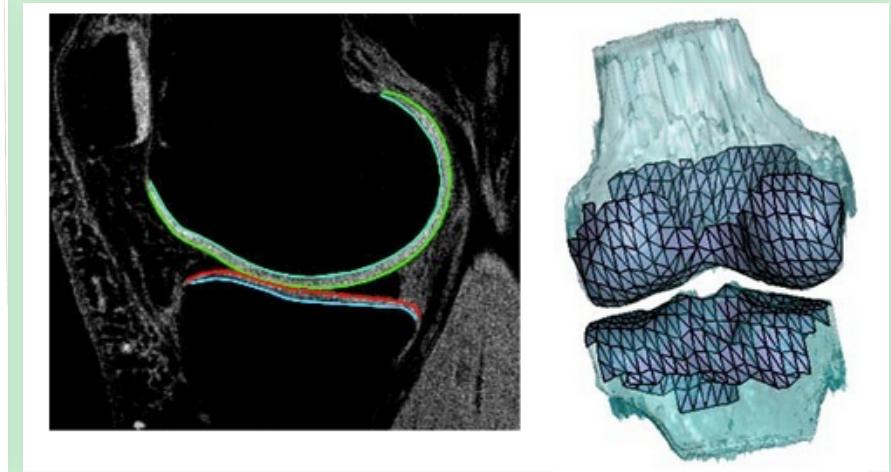
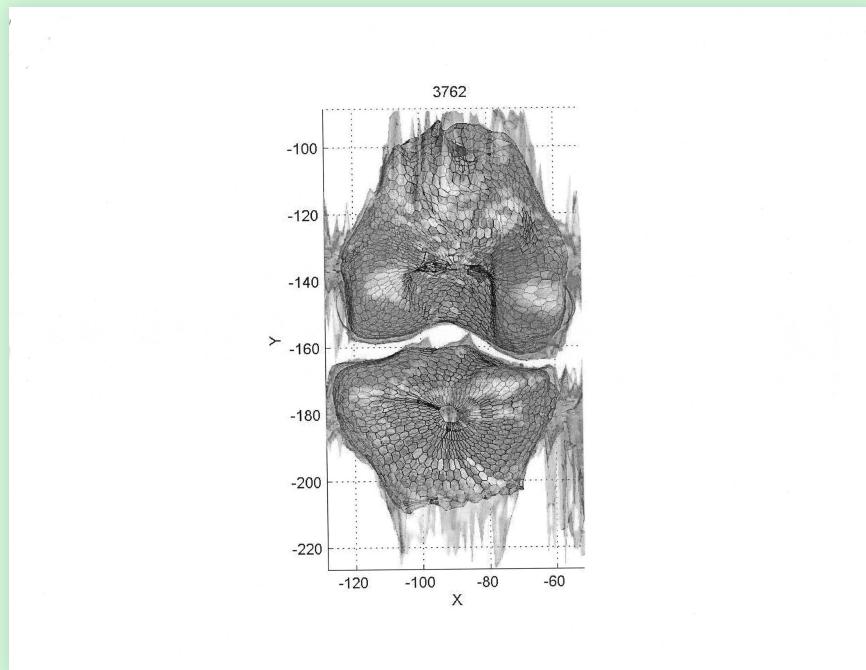
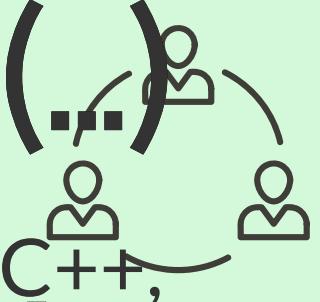
# PRÉSENTATION PERSONNELLE (...)

- 2008-2009: Capteur d'analyse réseau électrique, C++, simulink , Matlab



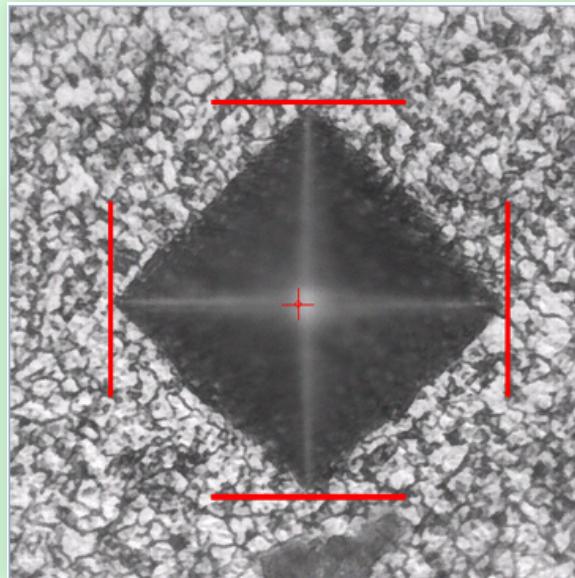
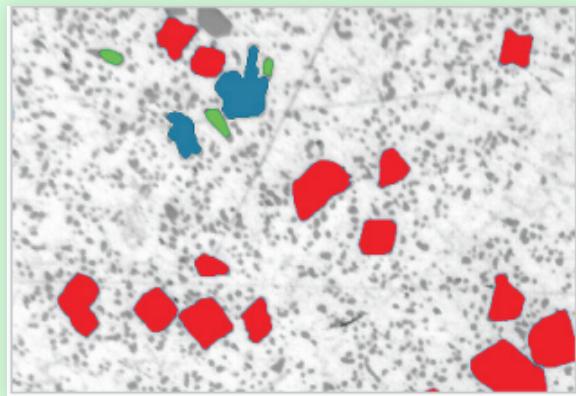
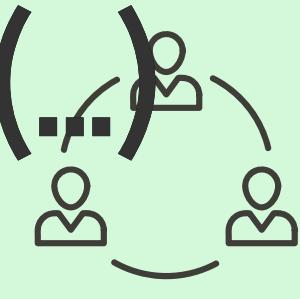
# PRÉSENTATION PERSONNELLE (...)

- 2001-2008: R&D en imagerie médicale, C++, Matlab



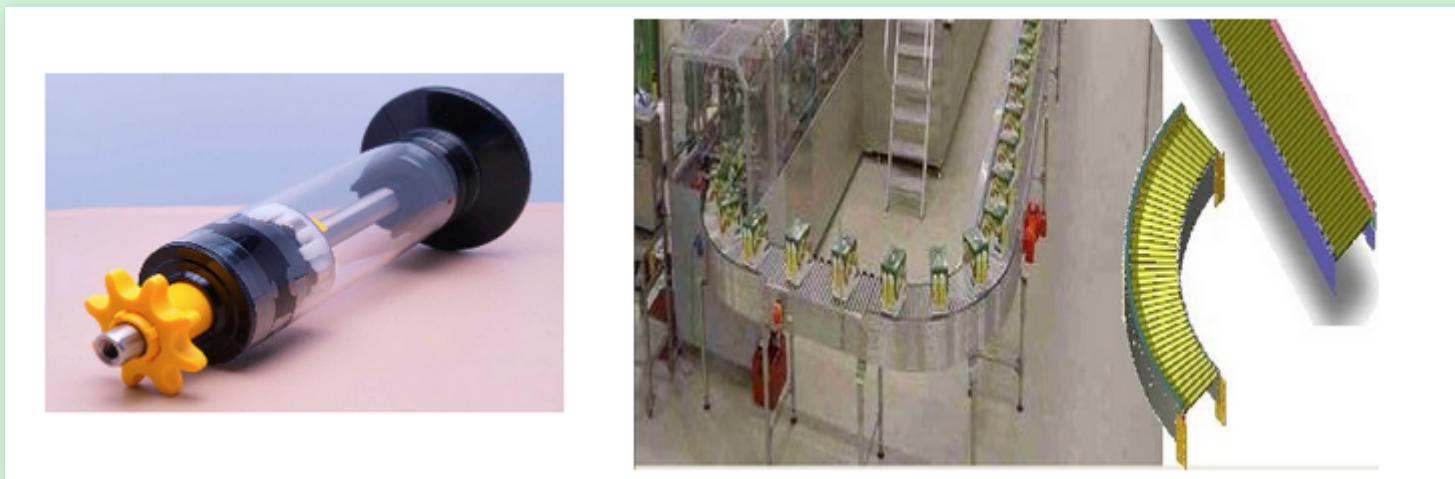
# PRÉSENTATION PERSONNELLE (...)

- 1998-2001: Développeur imagerie par microscopie, C++



# PRÉSENTATION PERSONNELLE (...)

- 1994-1998: Directeur R&D, conception de système d'inspection automatisée par vision, C++, conception d'équipement de convoyage

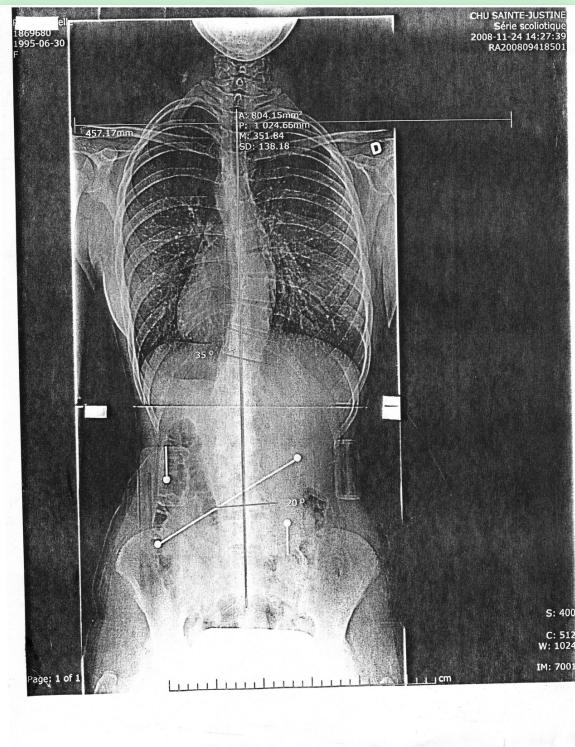


# PRÉSENTATION PERSONNELLE (...)

## DIPLOMES UNIVERSITAIRES



- 1994 Maîtrise en technologie des systèmes



# DIPLÔMES UNIVERSITAIRES

- 1990 Baccalauréat en Génie de la production automatisée



# LOG210 SÉANCE #01

## ANALYSE ET CONCEPTION DE LOGICIELS



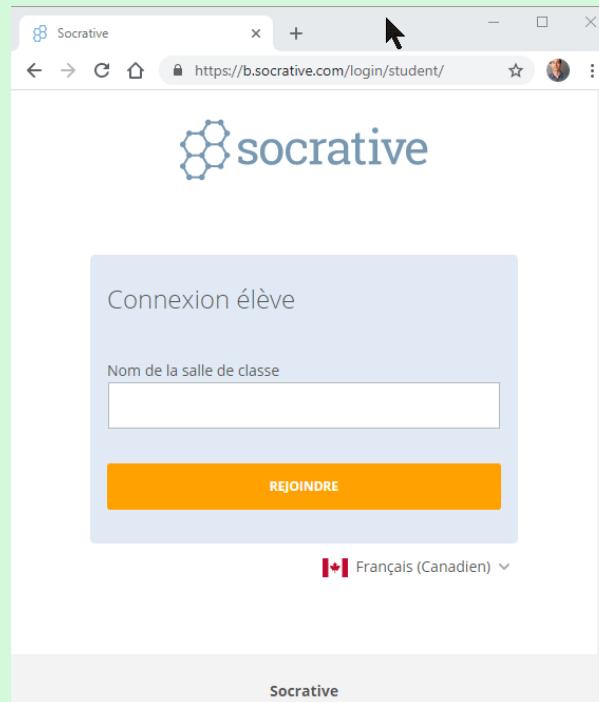
1. Présentation de l'équipe
2. Présentation personnelle
3. Présentations des étudiants 
4. Faire un logiciel pour un domaine que vous ne connaissez pas
5. Outils de travail
6. Introduction à l'analyse et conception de logiciels

# PRÉSENTATIONS DES ÉTUDIANTS



Questionnaire à [tiny.cc/quizdesign](https://tiny.cc/quizdesign)

Nom de la salle de class: **ETSDESIGN**



# PRÉSENTATIONS DES ÉTUDIANTS



- Comment développer un logiciel?
  - Quelle(s) méthode(s) de développement avez-vous suivie(s)?
  - Pour quel genre de logiciel?
- Quelles sont vos attentes pour LOG210?

# PRÉSENTATIONS DES ÉTUDIANTS

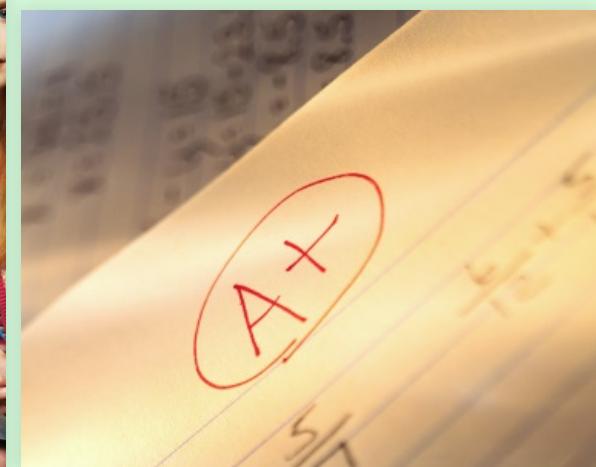


- Veuillez vous présenter sur Discord dans Général.
  - Votre nom, votre programme d'étude collégiale, votre programme d'étude universitaire, vos stages (entreprise, domaine, projet), pourquoi vous êtes en génie logiciel.



# MEILLEURES PRATIQUES POUR APPRENDRE

Quelles sont les meilleures pratiques pour apprendre?



# MES CONSEILS POUR RÉUSSIR



- L'étudiant qui réussit dans mes cours...
  - ~~s'assied vers l'avant de la salle,~~
  - est motivé pour apprendre et pour participer,
  - est bien discipliné,
  - est organisé,
  - cherche de l'aide d'une façon proactive,
  - sais que le **multitâche** nuit à la qualité.

# PROACTIF



1. **Domaine:** psychologie
2. **Auteur:** Office québécois de la langue française, 2002
3. **Définition:** Qui exerce un effet sur des faits ou des processus à venir

*Exemple: Pompiers qui visitent les maisons pour vérifier les détecteurs de fumée*



# INGÉNIEUR LOGICIEL, LE MEILLEUR MÉTIER AU MONDE ?



# INGÉNIEUR LOGICIEL, LE MEILLEUR MÉTIER AU MONDE ?



**Revers de la médaille:** certains reprochent à leur métier la nécessité de se mettre régulièrement au niveau. Ils trouvent difficile de suivre les évolutions avec les langages et les nouvelles pratiques qui ne cessent d'apparaître ou de se modifier



# INGÉNIEUR LOGICIEL, LE MEILLEUR MÉTIER AU MONDE ?



<http://www.developpez.com/actu/26910/Ingenieur-logiciel-le-meilleur-metier-au-monde-Oui-selon-une-etude-signee-CareerCast-com/>

# INGÉNIEUR LOGICIEL, LE MEILLEUR MÉTIER AU MONDE ?



Autre inconvénient: la solitude, notamment pour les développeurs qui préfèrent travailler depuis chez eux. Un choix qui leur offre un emploi du temps flexible, mais qui les mène à passer de longues heures sans interagir avec le moindre collègue.

<http://www.developpez.com/actu/26910/Ingenieur-logiciel-le-meilleur-metier-au-monde-Oui-selon-une-etude-signee-CareerCast-com/>

# LOG210 SÉANCE #01

## ANALYSE ET CONCEPTION DE LOGICIELS



1. Présentation de l'équipe
2. Présentation personnelle
3. Présentations des étudiants
4. Faire un logiciel pour un domaine que vous ne connaissez pas 
5. Outils de travail
6. Introduction à l'analyse et conception de logiciels

# FAIRE UN LOGICIEL POUR UN DOMAINE QUE VOUS NE CONNAISSEZ PAS

Exemple:

Application Web permettant de planifier les formations (faites par des spécialistes en TI) des enseignants dans les écoles québécoises.

Comment procéder?

- Appliquer un processus pour la réalisation
  - Appliquer une méthodologie d'analyse
  - Appliquer une méthodologie de conception



# LOG210 SÉANCE #01



## ANALYSE ET CONCEPTION DE LOGICIELS

1. Présentation de l'équipe
2. Présentation personnelle
3. Présentations des étudiants
4. Faire un logiciel pour un domaine que vous ne connaissez pas
5. Outils de travail 
6. Introduction à l'analyse et conception de logiciels



# OUTILS DE TRAVAIL



Created by  
From the Word Project

- Plan de cours
- Moodle - classe inversé
- Examen - enaquiz
- Google Drive pour le contenu
- Socrative
- Zoom / obs / youtube / discord
- one note
- laboratoires



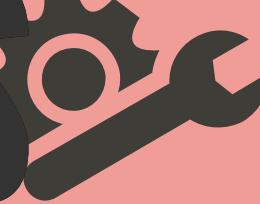
# OUTILS DE TRAVAIL...



- Visual studio code
  - extension
    - <https://visualstudio.microsoft.com/services/share/>
    - plantuml
    - Markdown all in One
    - Markdown PDF
- <https://www.npmjs.com/package/gitinspector>
- <https://www.npmjs.com/package/tplant>
- <https://plantuml.com/>
- <https://wakatime.com/dashboard>



# NOTES DE COURS



Created by  
From the Read Project

Il existe des notes de cours depuis janvier 2020.

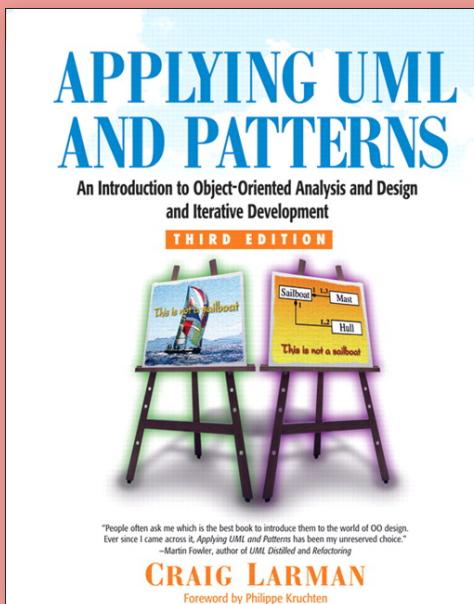
- [https://drive.google.com/file/d/1\\_yjzv7vuqbXOElxLj5AkVO0xEk7ubVO/view](https://drive.google.com/file/d/1_yjzv7vuqbXOElxLj5AkVO0xEk7ubVO/view)
- Les notes de cours ne remplacent pas le livre obligatoire



# LIVRE OBLIGATOIRE

Created by David  
From the Read Project

- Version anglaise ou française?



# LOG210 SÉANCE #01



## ANALYSE ET CONCEPTION DE LOGICIELS

1. Présentation de l'équipe
2. Présentation personnelle
3. Présentations des étudiants
4. Faire un logiciel pour un domaine que vous ne connaissez pas
5. Outils de travail
6. Introduction à l'analyse et conception de logiciels



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS...

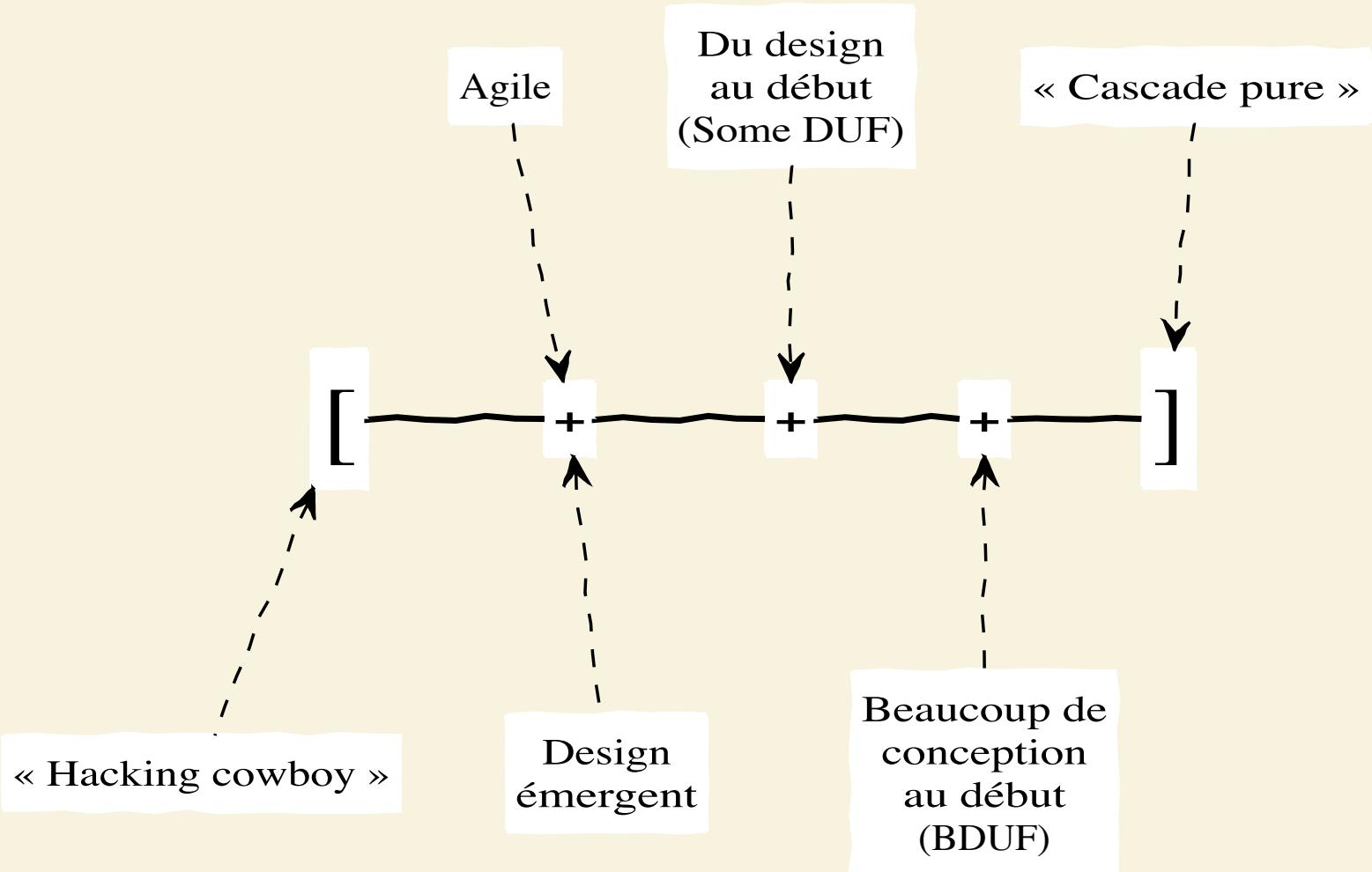
1. Spectre de la conception - voir 2
2. **Analyse et conception orienté objet**
3. **Equipe** - Travail d'équipe (HRC)
4. **PU** - Processus unifié
5. **DCU** - Diagramme de cas d'utilisation
6. **CU** - Cas d'utilisation
7. **IU** - Interface Usager vs DSS
8. **MDD** - Modèle du domaine
9. **DSS** - Diagramme de séquence système
10. **Contrats** - Contrats d'opération
11. **RDCU** - Réalisation d'un cas d'utilisation
12. **DCL** - Diagramme de classe logiciel



# SPECTRE DE LA CONCEPTION

Figure 2.1 des notes de cours.





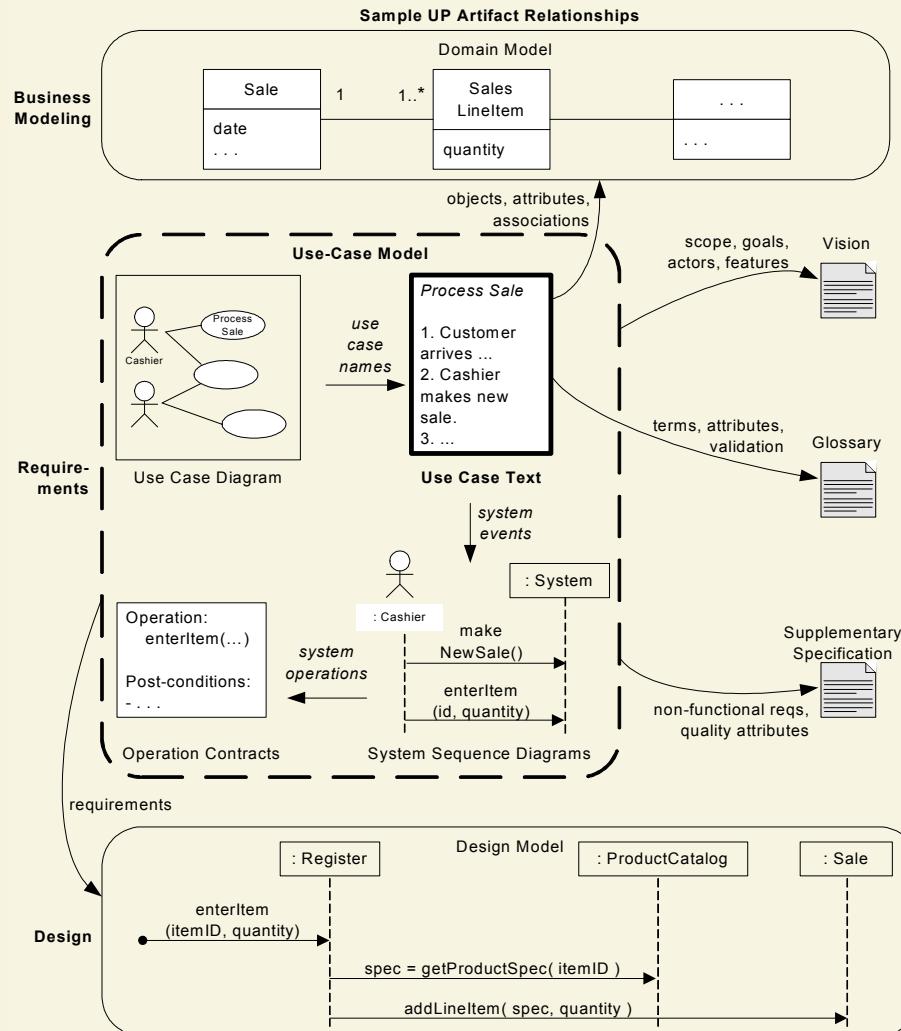
# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS

1. Spectre de la conception
2. Analyse et conception orienté objet  S20203
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



# SURVOL DES MODÈLES



# ANALYSE VS CONCEPTION

C'est de la modélisation!



# ANALYSE

- Faire un modèle du *problème*
  - Diagramme de classes conceptuelles
  - Modèle du domaine
- Investiger le problème
- Comprendre les besoins
- Ne pas (encore) proposer la solution



# CONCEPTION

- Faire un *modèle* d'une solution
  - Diagrammes de classes logicielles
  - Diagrammes d'interaction (séquence)
  - etc.
- Ce n'est pas la solution
- Facilite une évaluation des qualités comme la modularité



# DÉCALAGE DES REPRÉSENTATIONS

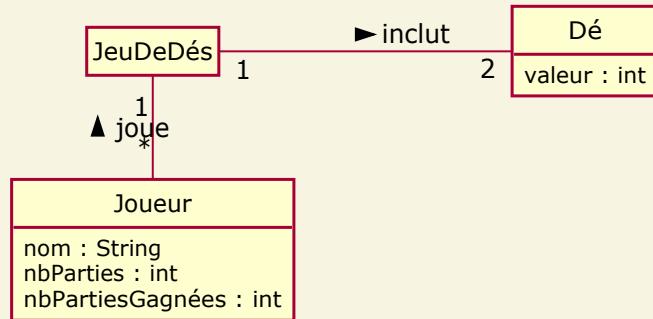
Comprenez-vous ce que fait ce programme?

```
00011101100010010010010100001011110011
```



# DÉCALAGE DES REPRÉSENTATIONS

Comprenez-vous ce que fait ce programme?



# DÉCALAGE DES REPRÉSENTATIONS

- Plus une solution (conception) ressemble à une description du problème, plus elle est facile à comprendre.
- La “distance” entre la représentation d'un problème et la représentation de sa solution s'appelle **le décalage des représentations**.

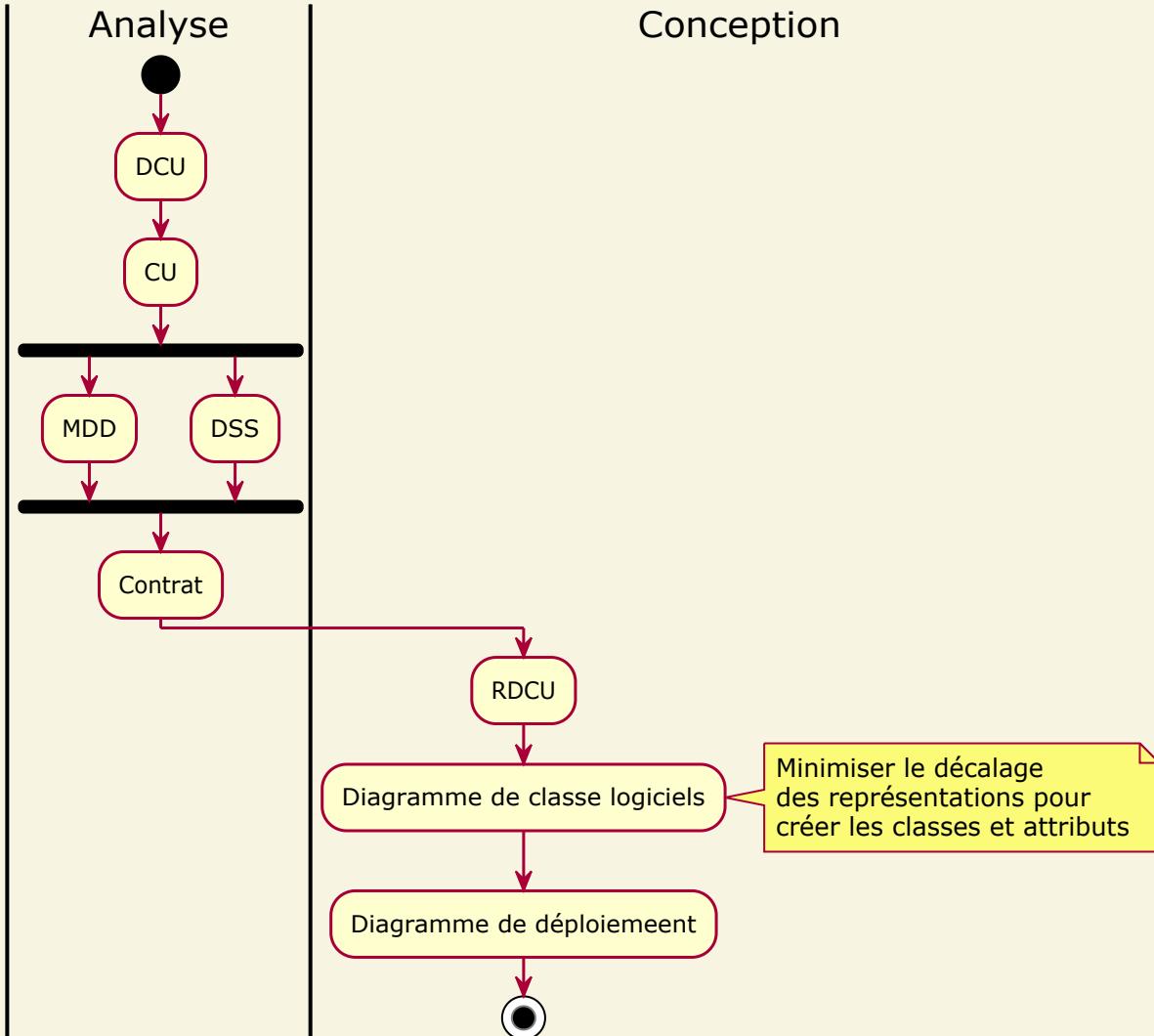


# ANALYSE VERSUS CONCEPTION



# Processus d'analyse et de conception de logiciels de Larman

---



# LECTURES À FAIRE

Le chapitre 1 des notes de cours.



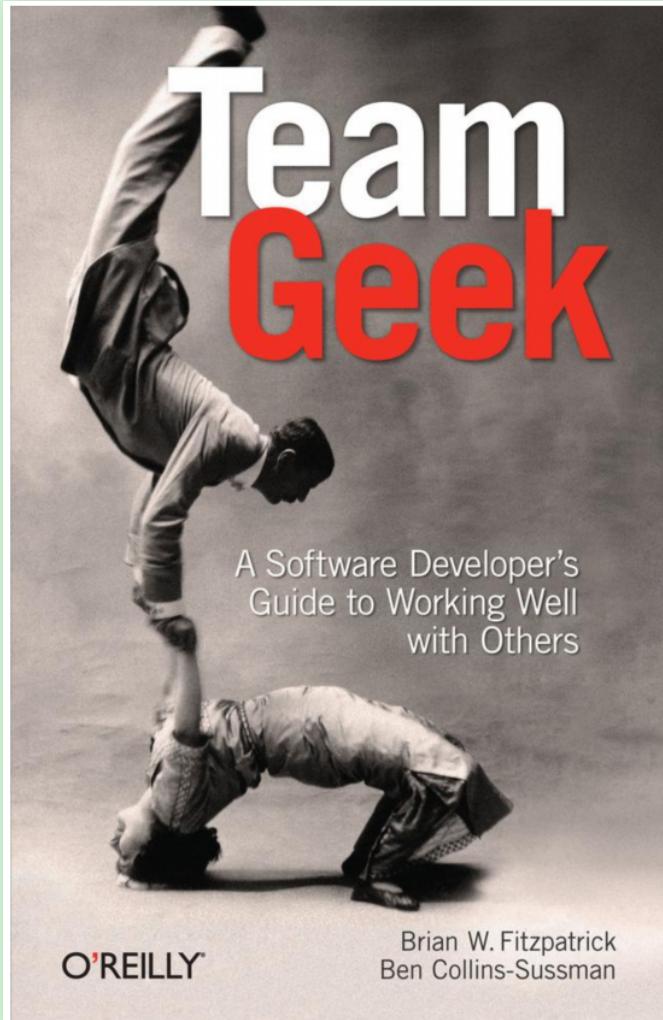
# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC) ← S20203
4. PU - Processus uniifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel

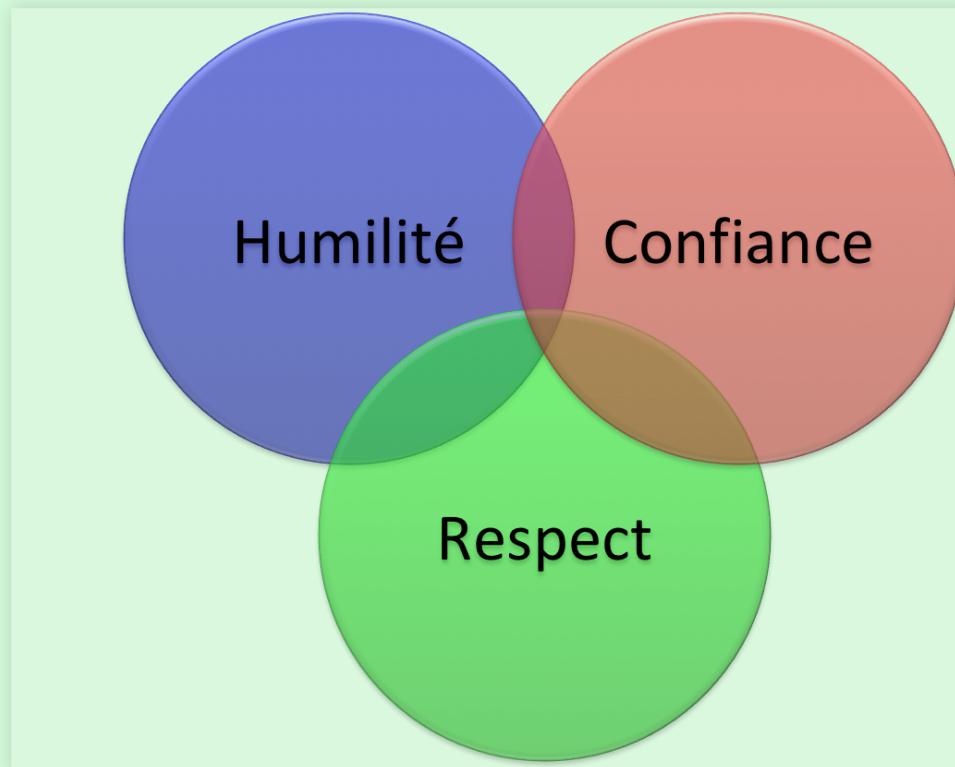
# DÉVELOPPEMENT DE LOGICIEL EN ÉQUIPE



# DÉVELOPPEMENT DE LOGICIEL EN ÉQUIPE



Pratiquement tout conflit social est dû à un ~~manque~~ d'humilité, de respect ou de confiance.





# RESPECT



Trouvez-moi un exemple d'un manque de respect:

- envers un coéquipier

# RESPECT



Trouvez-moi un exemple d'un manque de respect:

- envers un coéquipier
- envers les élèves de la classe

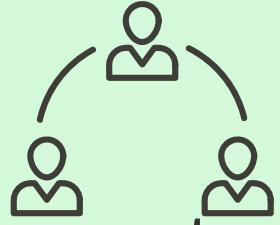
# RESPECT



Trouvez-moi un exemple d'un manque de respect:

- envers un coéquipier
- envers les élèves de la classe
- d'un étudiant envers un professeur ou un chargé de laboratoire

# RESPECT



Trouvez-moi un exemple d'un manque de respect:

- envers un coéquipier
- envers les élèves de la classe
- d'un étudiant envers un professeur ou un chargé de laboratoire
- d'un professeur/chargé de laboratoire envers un étudiant

# CONFiance



Trouvez-moi un exemple d'une:

- action qui donne confiance à mes coéquipiers

# CONFiance



Trouvez-moi un exemple d'une:

- action qui donne confiance à mes coéquipiers
- action qui diminue la confiance envers mes coéquipiers

# HUMILITÉ



Trouvez-moi un exemple ou:

- un étudiant ou un professeur fait preuve d'humilité

# HUMILITÉ



Trouvez-moi un exemple ou:

- un étudiant ou un professeur fait preuve d'humilité
- un étudiant ou un professeur ne fait pas preuve d'humilité

# POURQUOI L'ENSEIGNANT DÉCIDE LES MEMBRES DE L'ÉQUIPE? (1/2)



- Pour apprendre les compétences non-techniques
  - Demandées par les employeurs
  - N'évoluent pas autant que les compétences techniques (p.ex. Flash)
- On ne choisit pas (facilement) son équipe en entreprise
- Travailler avec des collègues différents tout au long de son baccalauréat élargit son réseau de contacts
- Impact positif pour la carrière

Mark Granovetter, «The Strength of Weak Ties », Academic Press, 1977.

# POURQUOI L'ENSEIGNANT DÉCIDE LES MEMBRES DE L'ÉQUIPE? (2/2)



- Favorise le « coaching » car il y a de la diversité
- Un coéquipier moins fort dans un aspect du projet peut bénéficier du coaching d'un membre plus fort
- Le « coach » peut bénéficier également de ce qu'on appelle «deep learning»
- L'équipe est finalement plus cohésive et plus performante

# ÉVITER LES COMPORTEMENTS DYSFONCTIONNELS



- Chacun fait “sa part” sans se soucier de la santé de l’équipe et de la qualité globale du projet
  - L’équipe > moi (appliquer l’humilité)
  - Rester flexible pour les tâches, s’adapter
  - Proactivité envers problèmes dans l’équipe:
    - Exiger la présense des coéquipiers
    - Demander de l’aide (technologies)
    - S’impliquer!

# ÉVITER LES COMPORTEMENTS DYSFONCTIONNELS

- Un ou deux membres prennent tout le contrôle (manque de confiance)
  - Reconnaître les niveaux différents et aider (faire du coaching)
  - Être patient avec les membres ayant moins d'expérience
  - Membres ayant besoin d'aide doivent la chercher tôt (être proactif, s'impliquer)
  - Facteur de bus > 1

# LOG210 SÉANCE #01

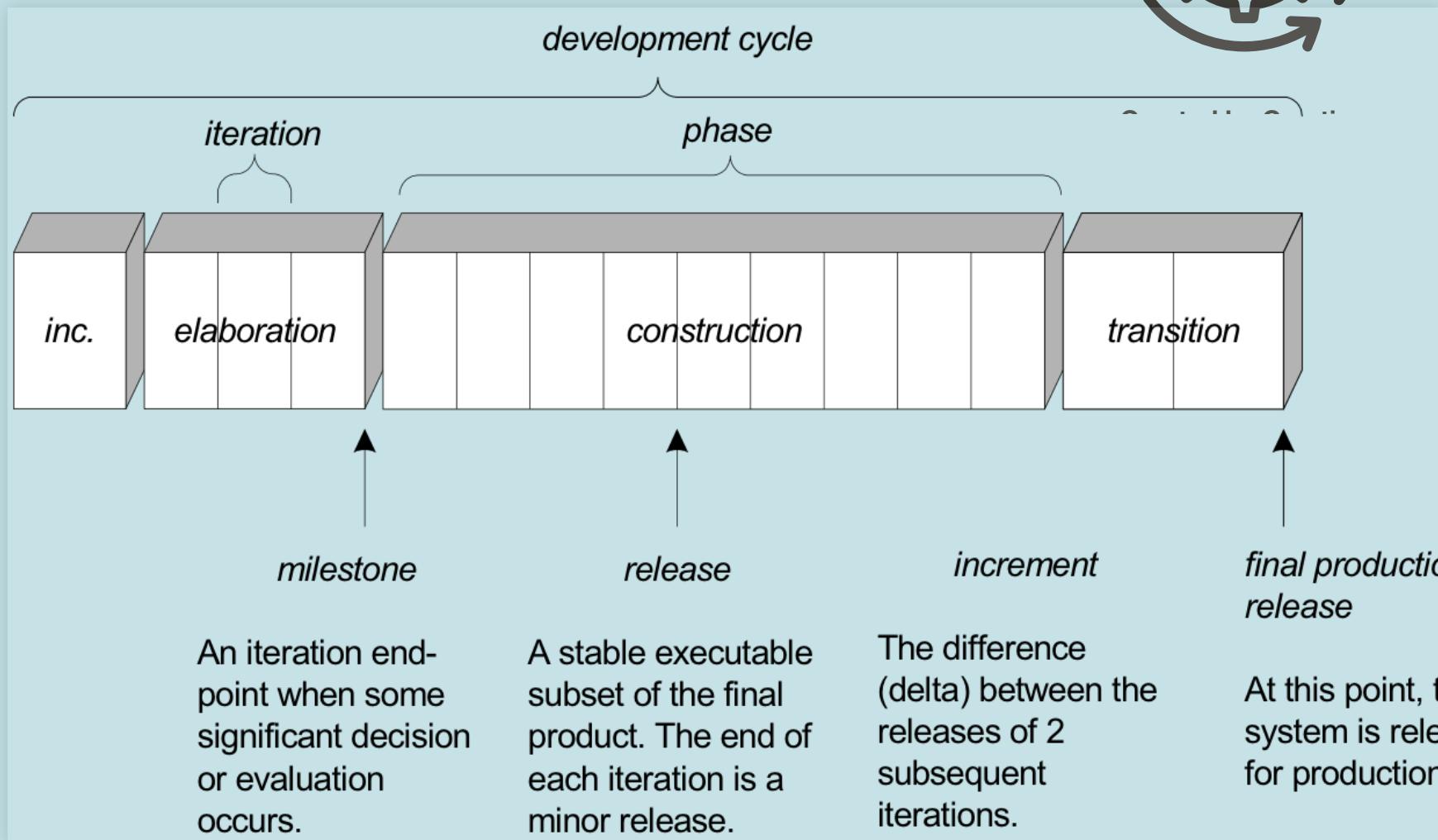
## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



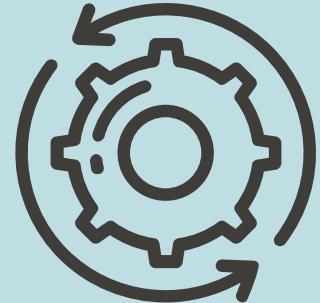
1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié ← S2003
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



# PHASES DU PROCESSUS UNIFIÉ



# PHASE INCEPTION



- Étude de faisabilité
  - Vision approximative
  - Estimations globales
  - Cas d'utilisation
  - Continuer ou non (p.ex. chercher le financement)



- Ne pas confondre avec
  - analyse des exigences



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS

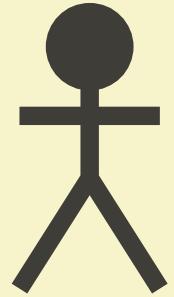


Created by Swen-Peter Ekkebus  
from the Noun Project

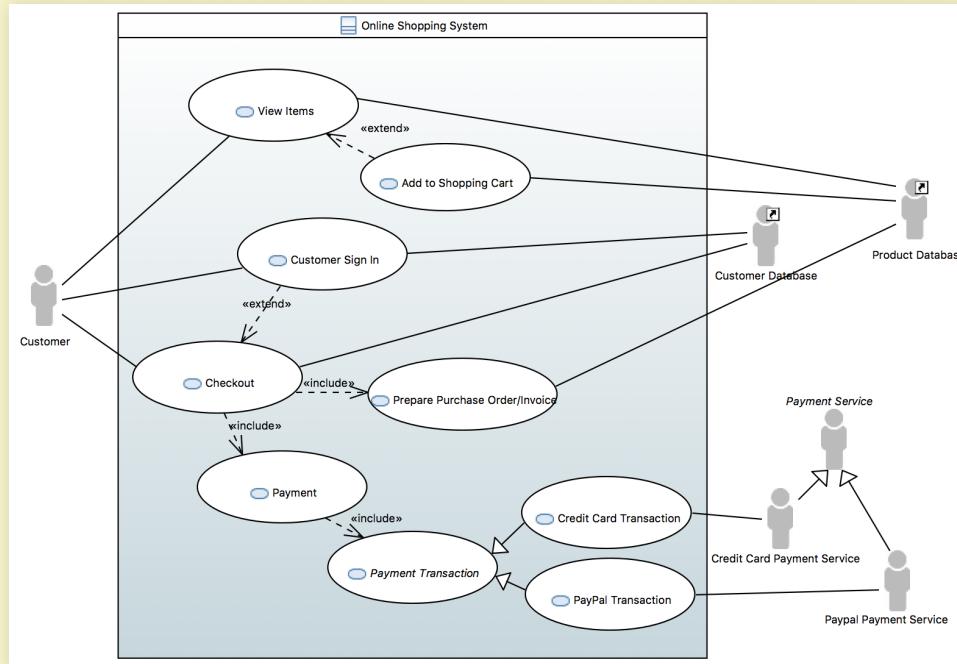
1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation  S20203
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel

# DCU (LOG410)

## DIAGRAMME DE CAS D'UTILISATION

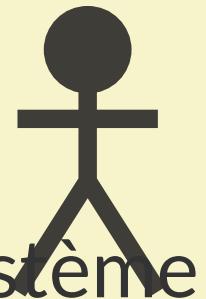


Created by Swen-Peter Ekkebus  
from the Noun Project



<https://plantuml.com/use-case-diagram>

# CONCEPTS DE BASE DU DIAGRAMME DE CAS D'UTILISATION



## 1. Sujet (subject)

- Le sujet est utilisé pour délimiter le système de son environnement

## 2. Acteur

- Entité externe qui interagit avec le système
- Peut représenter des personnes / utilisateurs et d'autres systèmes externes avec lesquels le système interagit

Created by Swen-Peter Ekkebus  
from the Noun Project



Created by Swen-Peter Ekkelbus  
from the Noun Project

## 1. Cas d'utilisation

- Fonctionnalité fournie par le système
- Décrit comme une séquence d'étapes / actions

## 2. Relations

- L'interaction entre les acteurs et un cas d'utilisation est spécifiée à l'aide d'une association.
- Différents types de relations entre les cas d'utilisation: include, extend, generalization/inheritance

# TROIS TYPES D'ACTEURS



Created by Swen-Peter Ekkelbus  
from the Noun Project

## 1. Acteur principal

- il a un objectif et utilise le système à l'étude (**SAE**)

## 2. Acteur de support

- fournit un service, une information, au SAE
- p.ex. système d'autorisation de paiement. Peut être humain, système externe ou équipement.

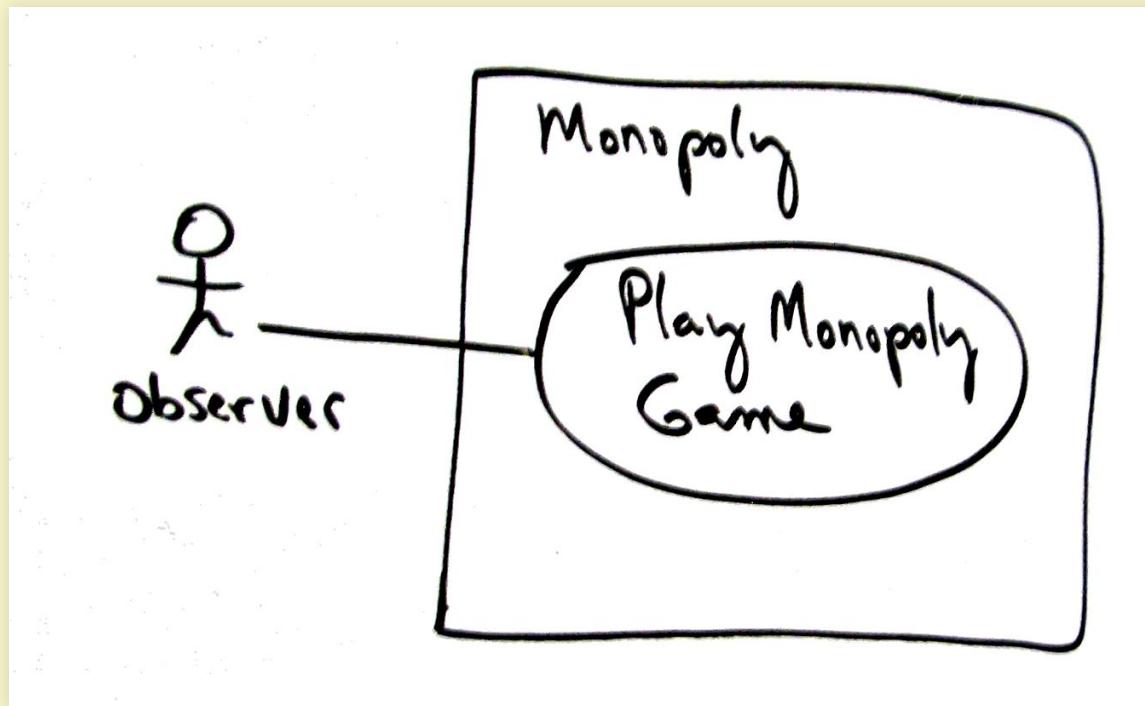
## 3. Intervenant

- a un intérêt, mais ni acteur principal ou acteur de support (ex: service des taxes du gouvernement, comptable, vérificateur, propriétaire, client, ...)

# DIAGRAMME DE CAS D'UTILISATION (AGILE)



Created by Swen-Peter Ekkebus  
from the Noun Project



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



Created by Adrien Coquet  
from the Noun Project

1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation  S20203
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



CU



Created by Adrien Coquet  
from the Noun Project

# CAS D'UTILISATION



# CAS D'UTILISATION - INFORMEL



**1. Traiter une vente.** Un client arrive à la caisse avec les articles qu'il souhaite acheter. Pour enregistrer chaque article, le caissier utilise le système POS, lequel présente le détail des articles et le montant total des achats. Le client fournit les informations nécessaires pour le règlement. Le système valide et enregistre ces informations, puis met à jour les quantités en stock et imprime le ticket de caisse destiné au client. La vente est terminée et le client peut quitter le magasin.

Created by Adrien Coquet  
from the Noun Project



# DE LA DESCRIPTION DU CAS D'UTILISATION AU DIAGRAMME DE CAS D'UTILISATION

## 4 Basic Flow of Events

1. Customer selects the "Checkout" option
2. If the Customer is not already signed in, then Cmind OSS ask the Customer to sign in – described in Subflow 6.1
3. Cmind OSS retrieves Customer information from Customer Database
4. Cmind OSS displays existing shipping address and asks Customer to confirm
5. Customer confirms shipping address or enter a different shipping address
6. Cmind OSS asks Customer to select shipping option
7. Customer chooses a shipping option
8. Cmind OSS prepares purchase order/invoice – described in Subflow 6.2
9. Cmind OSS displays the final purchase order/invoice
10. Cmind OSS proceeds with payment – described in Subflow 6.3
11. Cmind OSS displays final transaction receipt and send a copy to Customer email address
12. Cmind OSS returns to the Welcome page

## 6 Subflows

### 6.1 Customer Sign In – described by the Customer Sign In use case

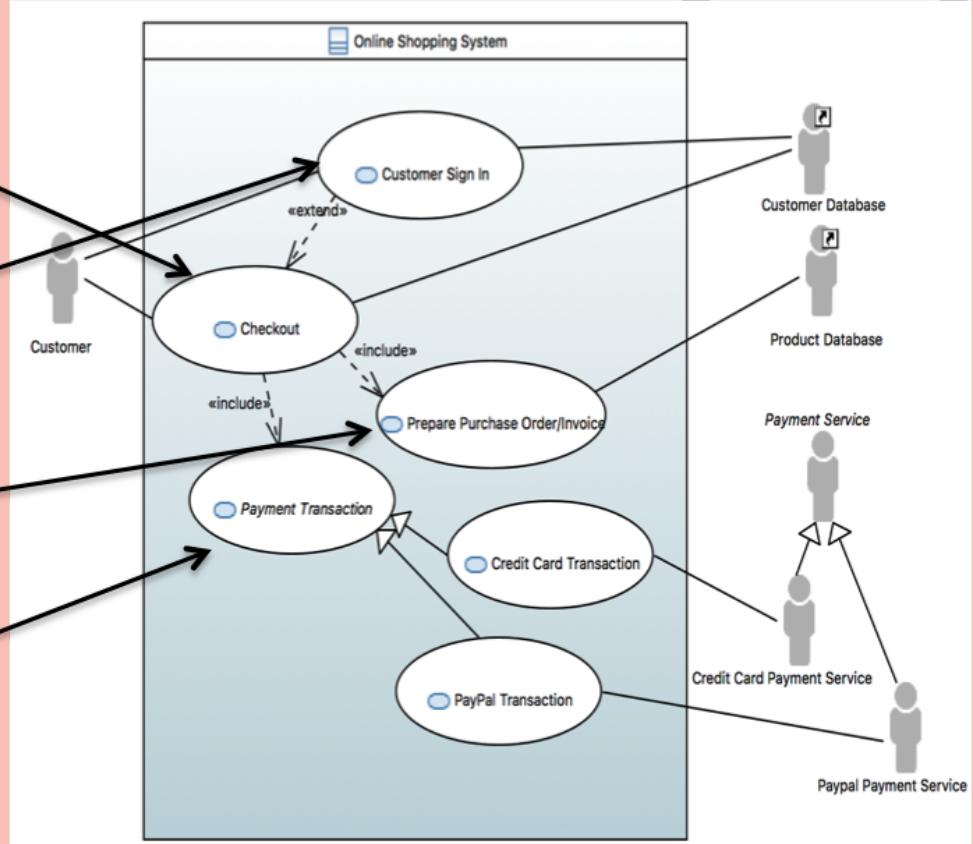
1. Cmind OSS asked Customer to enter username and password
  2. Customer enters username and password
  3. Cmind OSS validates Customer login information with Customer Database
  4. Cmind OSS asks Customer Database to validate Customer Sign In information
- Alternative flow: If Customer login information is invalid, then the use case is terminated*

### 6.2 Prepare Purchase Order/Invoice – described by the Prepare Purchase Order/Invoice use case

1. Cmind OSS retrieves content of the shopping cart
2. Cmind OSS retrieves product information from Product Database
3. Cmind OSS calculates purchase sub-total (excluding taxes and shipping) based on content of shopping cart and product info
4. Cmind OSS adds shipping cost to sub-total
5. Cmind OSS calculates taxes
6. Cmind OSS calculates purchase total

### 6.3 Payment – described by the Payment use case

1. Cmind OSS asks Customer to select payment method – described by the Payment use case
2. Cmind OSS asks Customer to select payment method
3. Customer chooses the payment method
4. Cmind OSS displays the existing Customer payment information for the selected method and asks Customer to confirm information
5. Customer confirms payment information
6. Customer makes final purchase confirmation
7. Cmind OSS processes the payment transaction – described by Payment Transaction use case



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS

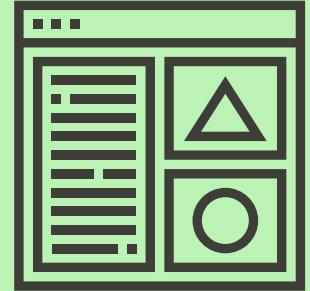


Created by Eucalyp

1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS ← S20203
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



IU

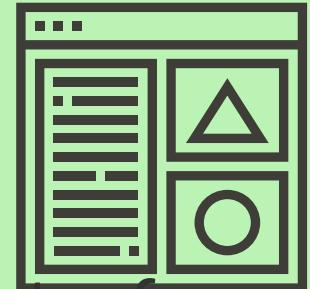


Created by Eucalyp

# INTERFACE USAGER



# INTERFACE USAGER

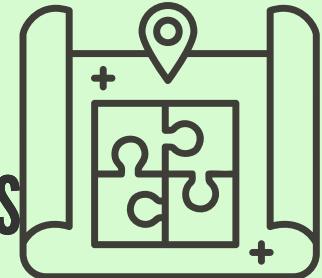


Created by Eucalyp

- Réaliser les wireframes des différentes interfaces usagées nécessaires pour la réalisation du cas d'utilisation **Noter une réservation**
- Utiliser plant uml
  - <https://plantuml.com/salt>

# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine  S20203
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



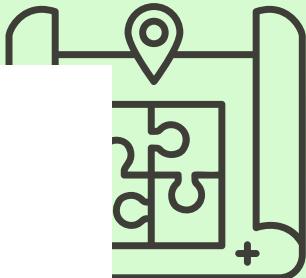
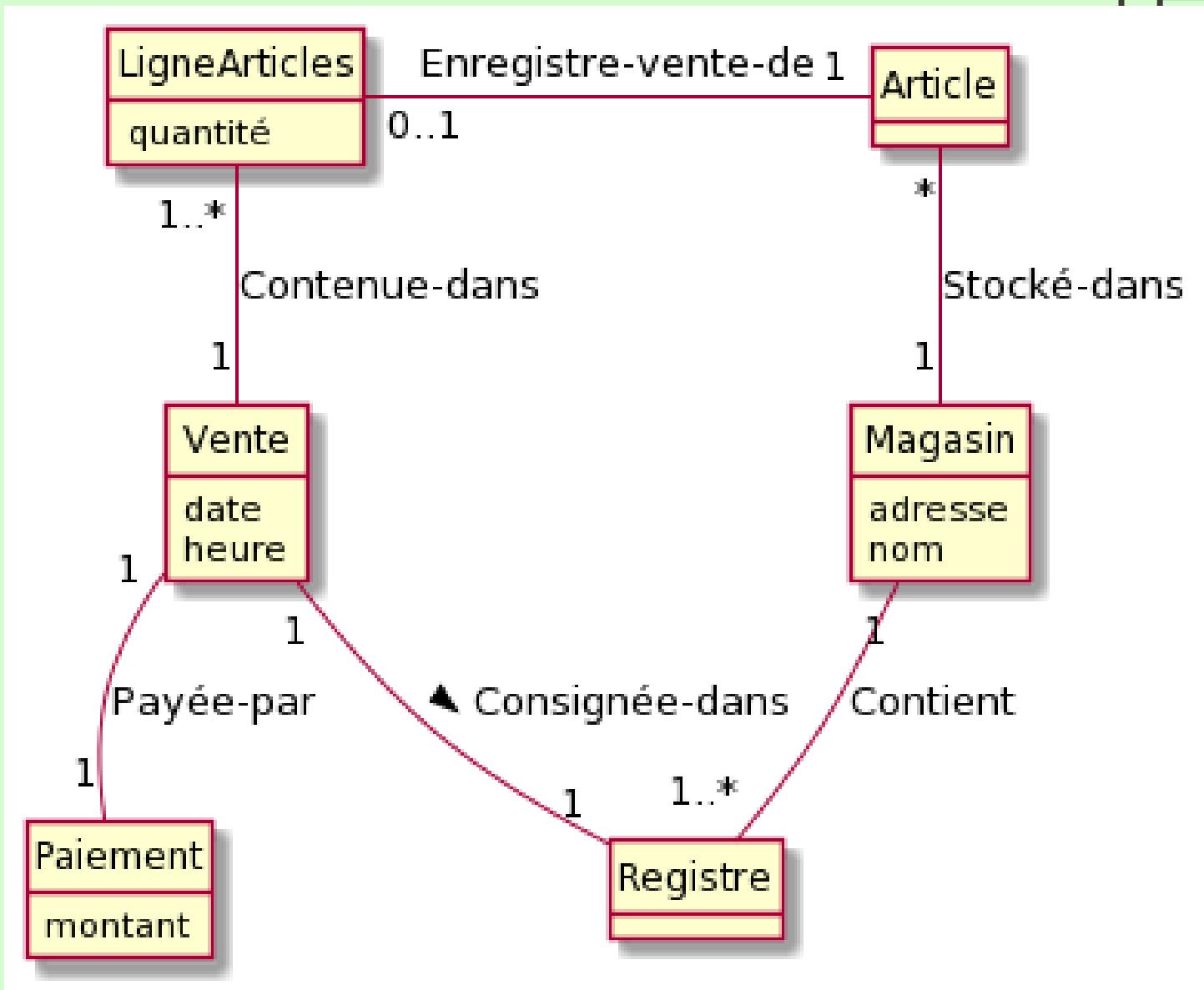
# QU'EST-CE QU'UN MODÈLE DU DOMAINE?



Représentation visuelle des classes conceptuelles ou des objets du monde réel dans un domaine donné  
[MO95, Fowler96]

- modèle conceptuel
- modèle objet du domaine
- modèle objet d'analyse

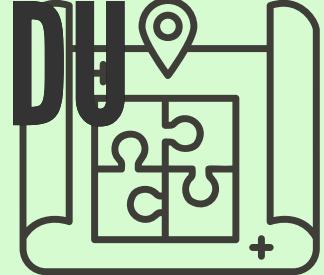
# EXEMPLE AVEC L'OUTIL PLANTUML



# Plant Uml class diagram



# COMMENT CRÉER UN MODÈLE DU DOMAINE?



- Dans les limites des besoins de l’itération en cours:
  1. Identifier les classes conceptuelles.
  2. Les représenter sous forme de classes dans un diagramme UML.
  3. Ajouter des attributs et des associations.

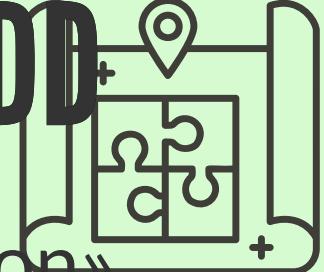
# RÉALISER VOTRE PREMIER MDD



En équipe de 2

1. Trouver les classes du domaine
2. Ajouter des associations entre les classes lorsque vous désirez persister cette relation
3. Ajouter un verbe pour décrire cette association
4. Ajouter une multiplicité à l'association

# RÉALISER VOTRE PREMIER MDD



Cas d'utilisation «Noter une réservation»

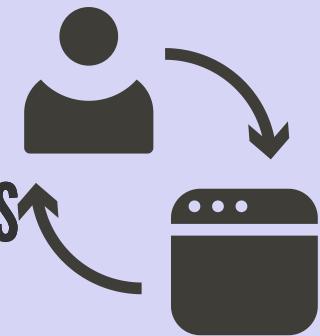
## Scénario principal

1. Un client appelle à l'hôtel pour placer une réservation.
2. Le commis démarre une nouvelle réservation.
3. Le commis entre:
  - La date d'arrivée;
  - La date de départ;
  - Le nom de la catégorie de chambre;
4. Le système inscrit les informations à la réservation.

Interface usager | Réaliser votre premier DSS | Itération suivante

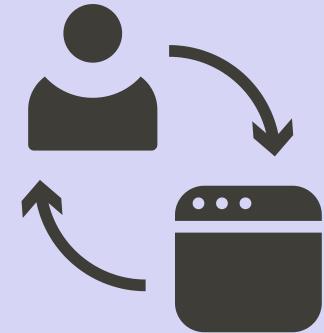
# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



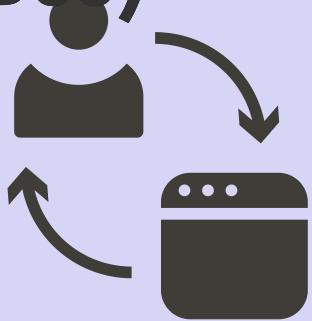
1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système ← S20203
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel

# QU'EST-CE QU'UN DSS?



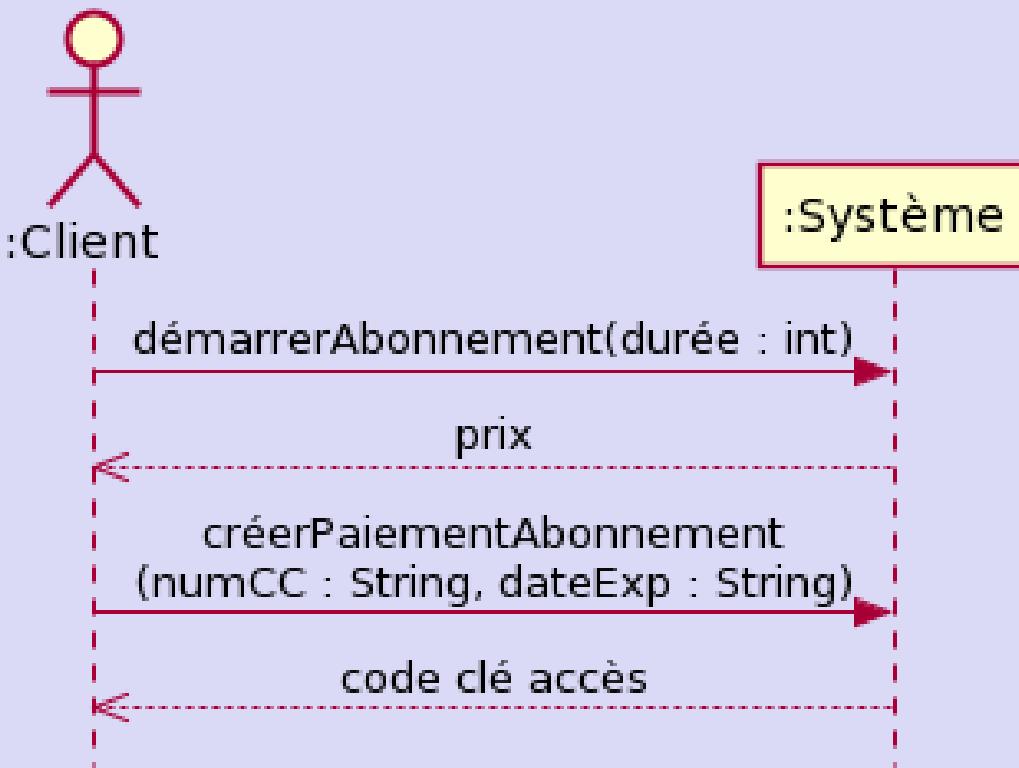
- Interaction avec le système
  - acteur génère des événements
  - réponses du système
- Illustrer ces opérations
  - pour mieux comprendre le système

# DIAGRAMME DE SÉQUENCE DU SYSTÈME (DSS)



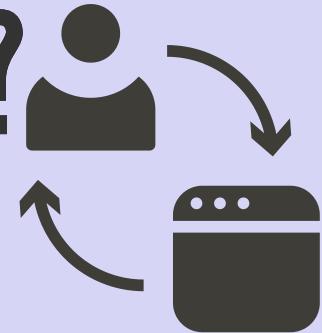
- événements synchrones
- paramètres
- retour d'information
- frontières du système

## Traiter abonnement



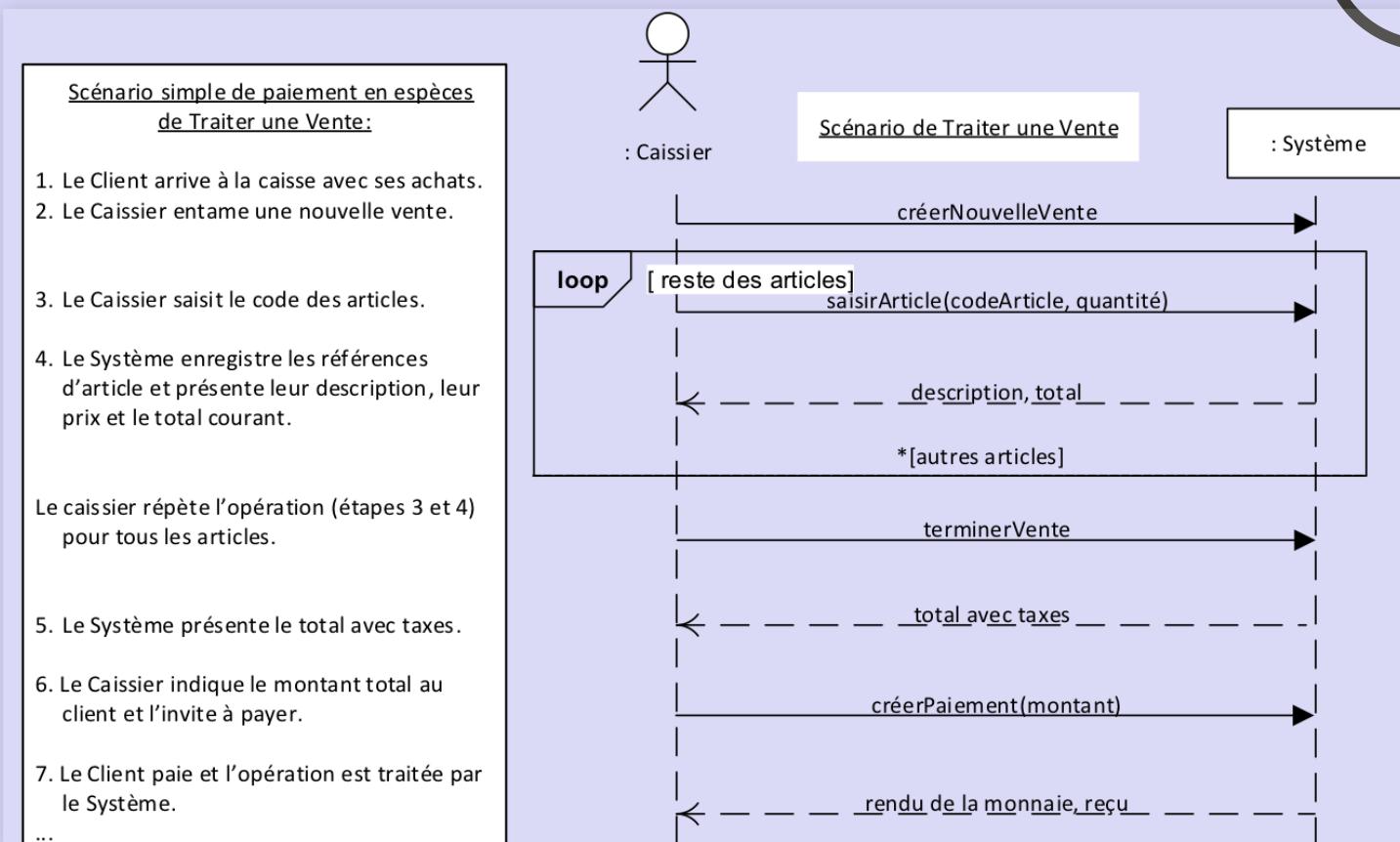
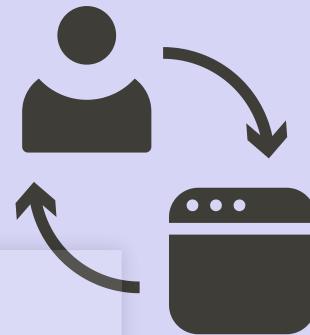


# POURQUOI TRACER UN DSS?



- Pour connaître les événements système
  1. déclenchés par acteurs (humains ou ordis)
  2. d'échéance de temporisation (timeout)
  3. d'erreurs ou d'exceptions (sources externes)
- Début de la modélisation par décomposition (top-down)
- Vu abstrait du système comme application
- Facilite le changement de l'interface humain-machine (la couche présentation)

# DSS ET CAS D'UTILISATION



<https://plantuml.com/sequence-diagram>

# RÉALISER VOTRE PREMIER DSS

- Réaliser un diagramme de séquence système du cas d'utilisation Noter une réservation.



# IU - INTERFACE USAGER VS DSS

Le DSS peut vous permettre de définir les interfaces usager nécessaires pour appeler les opérations systèmes et afficher les retours d'informations.

- Vérifier la correspondance entre vos interfaces usager et vos opérations systèmes



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



Created by Mohammed Ra

1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération ← S20203
11. RDCU - Réalisation d'un cas d'utilisation
12. DCL - Diagramme de classe logiciel



# Contrats



- Transformation au niveau des données dans le MDD suite à l'exécution d'une opération du DSS

Created by Mohammed Rai

# Contrats



Created by Mohammed Rai

- Transformation au niveau des données dans le MDD suite à l'exécution d'une opération du DSS
- Expliquer en vos propre mots se qui ce passe au niveau des données de votre modèle du domaine lorsque vous appelez chacune des opérations système.

# Contrats



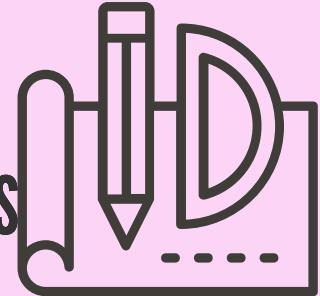
Created by Mohammed Rai

- Transformation au niveau des données dans le MDD suite à l'exécution d'une opération du DSS
- Expliquer en vos propre mots se qui ce passe au niveau des données de votre modèle du domaine lorsque vous appelez chacune des opérations système.
- Opération système == contrat



# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



1. Spectre de la conception
2. Analyse et conception orienté objet
3. Equipe - Travail d'équipe (HRC)
4. PU - Processus unifié
5. DCU - Diagramme de cas d'utilisation
6. CU - Cas d'utilisation
7. IU - Interface Usager vs DSS
8. MDD - Modèle du domaine
9. DSS - Diagramme de séquence système
10. Contrats - Contrats d'opération
11. RDCU - Réalisation d'un cas d'utilisation ← s20203
12. DCL - Diagramme de classe logiciel



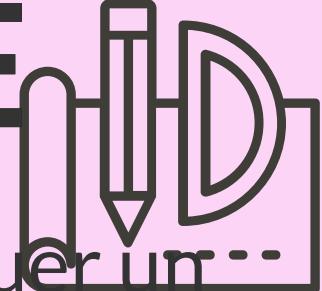
# RÉALISATION D'UN CAS D'UTILISATION



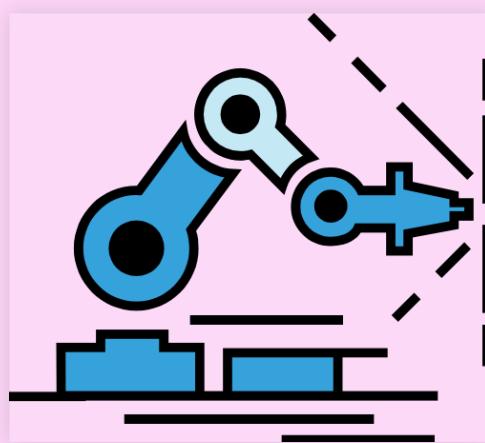
- Un diagramme d'interaction avec des annotations pour réaliser une opération système et satisfaire:
  - Son contrat
  - Ses retours d'informations
- Deux types de responsabilités
  - Faire
  - Savoir

<https://plantuml.com/sequence-diagram>

# RESPONSABILITÉS DE FAIRE

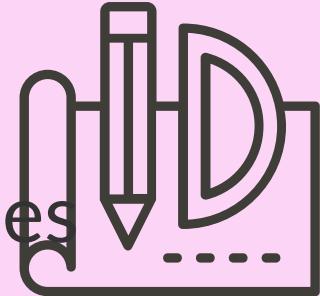


- faire quelque chose lui-même (ex. effectuer un calcul)
- déclencher une action d'un autre objet
- contrôler et coordonner les activités d'autres objets



# RESPONSABILITÉS DE SAVOIR

- connaître les données privées encapsulées
- connaître les objets connexes
- connaître des éléments qu'il peut dériver ou calculer



# LA CPR EST UNE MÉTAPHORE POUR LA COOPÉRATION



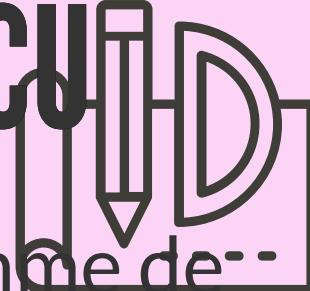
- On y assimile les objets à des personnes qui
  - ont des responsabilités
  - collaborent avec d'autres personnes
  - réalisent une tâche



CPR = Conception pilotée par les responsabilités

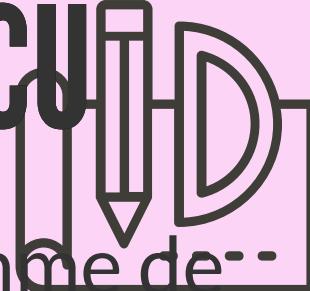


# RÉALISER VOS PREMIERS RDCU



- Concrètement: Un RDCU est un diagramme de séquence ou de communication qui réalise le contrat associé à l'opération système et s'assure de pouvoir retourner les informations demandées.

# RÉALISER VOS PREMIERS RDCU



- Concrètement: Un RDCU est un diagramme de séquence ou de communication qui réalise le contrat associé à l'opération système et s'assure de pouvoir retourner les informations demandées.
- Créer un contrôleur

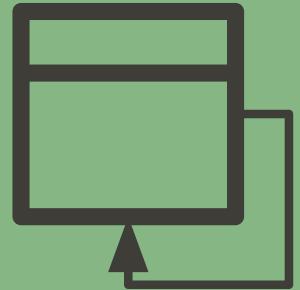
# RÉALISER VOS PREMIERS RDCU



- Concrètement: Un RDCU est un diagramme de séquence ou de communication qui réalise le contrat associé à l'opération système et s'assure de pouvoir retourner les informations demandées.
- Créer un contrôleur
- Faire un diagramme de séquence ou de communication pour chaque opération système.

# LOG210 SÉANCE #01

## INTRODUCTION À L'ANALYSE ET CONCEPTION DE LOGICIELS



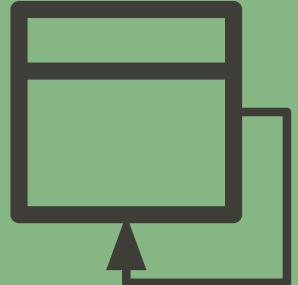
Created by Swen-Peter Ekkebus  
from the Noun Project

1. **Spectre de la conception**
2. **Analyse et conception orienté objet**
3. **Equipe** - Travail d'équipe (HRC)
4. **PU** - Processus uniifié
5. **DCU** - Diagramme de cas d'utilisation
6. **CU** - Cas d'utilisation
7. **IU** - Interface Usager vs DSS
8. **MDD** - Modèle du domaine
9. **DSS** - Diagramme de séquence système
10. **Contrats** - Contrats d'opération
11. **RDCU** - Réalisation d'un cas d'utilisation
12. **DCL** - Diagramme de classe logiciel

← s20203



# DCL - DCC

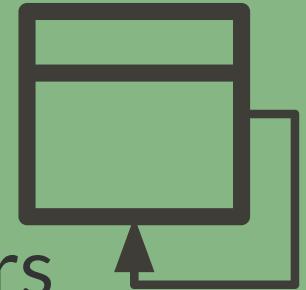


Created by Swen-Peter Ekkebus  
from the Noun Project

## DIAGRAMME DE CLASSE LOGICIEL

## DIAGRAMME DE CLASSE DE CONCEPTION

# CLASSIFICATEURS UML

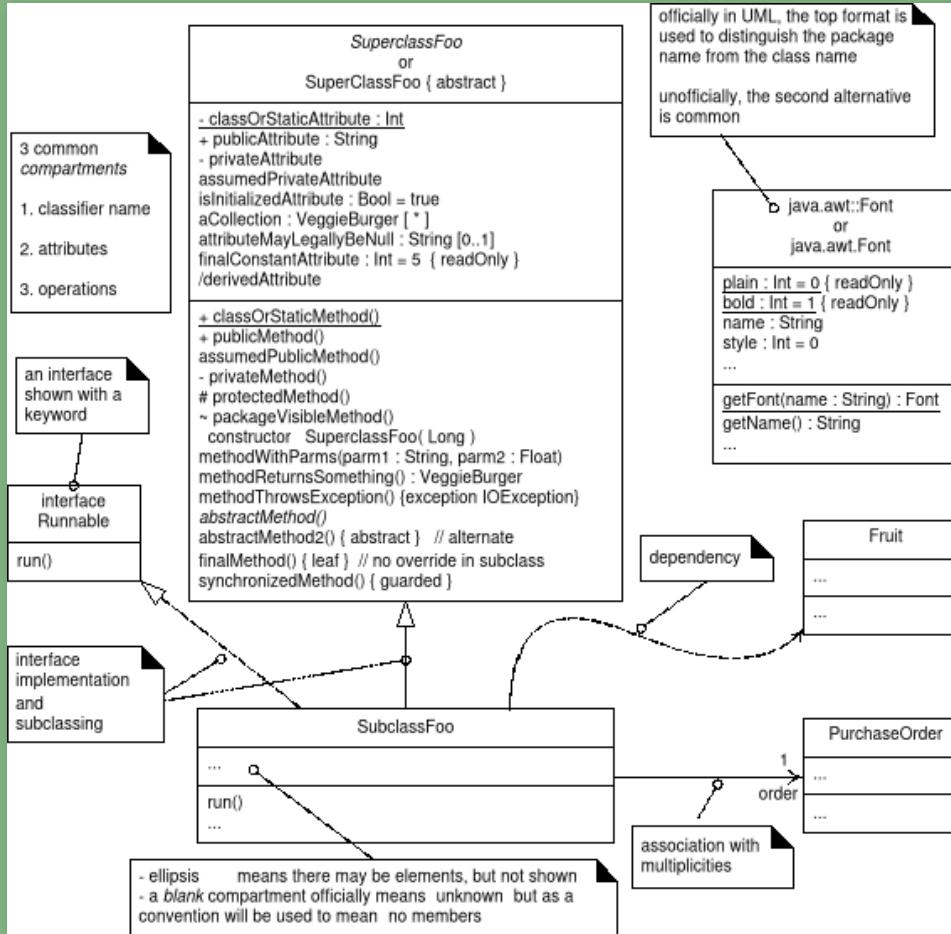


- Dans les DCC, il y en a deux classificateurs courants:
  - Les classes
  - Les interfaces

Created by Swen-Peter Ekkebus  
from the Noun Project

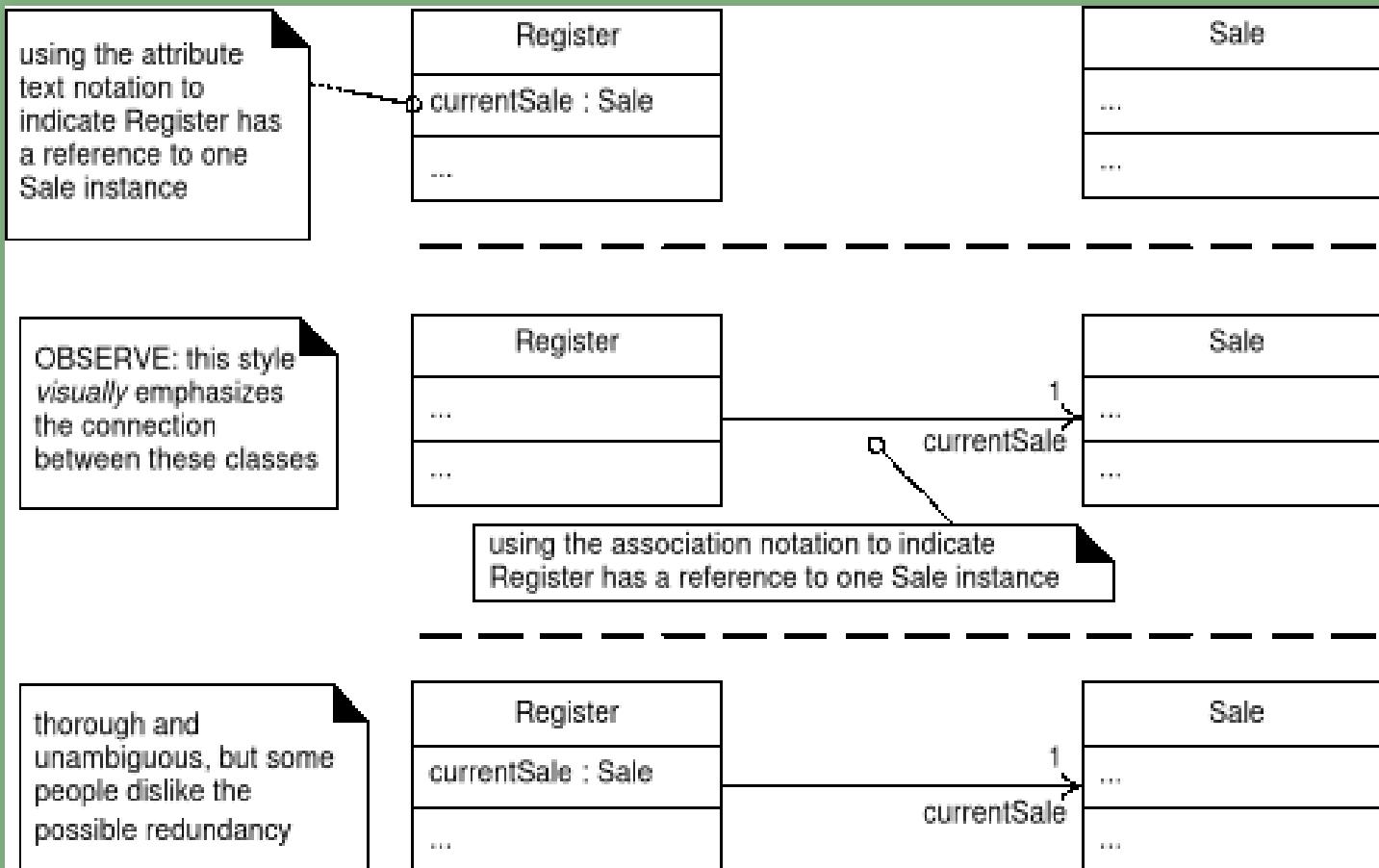
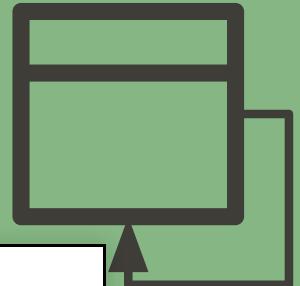
<https://plantuml.com/class-diagram>

# NOTATION COURANTE (VUE EN LOG210)



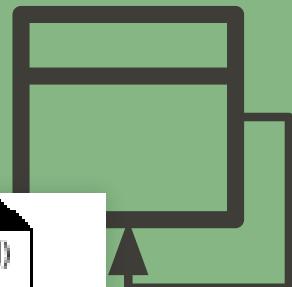
Created by Sven-Peter Ekkelbus  
from the Noun Project

# ATTRIBUTS EN UML

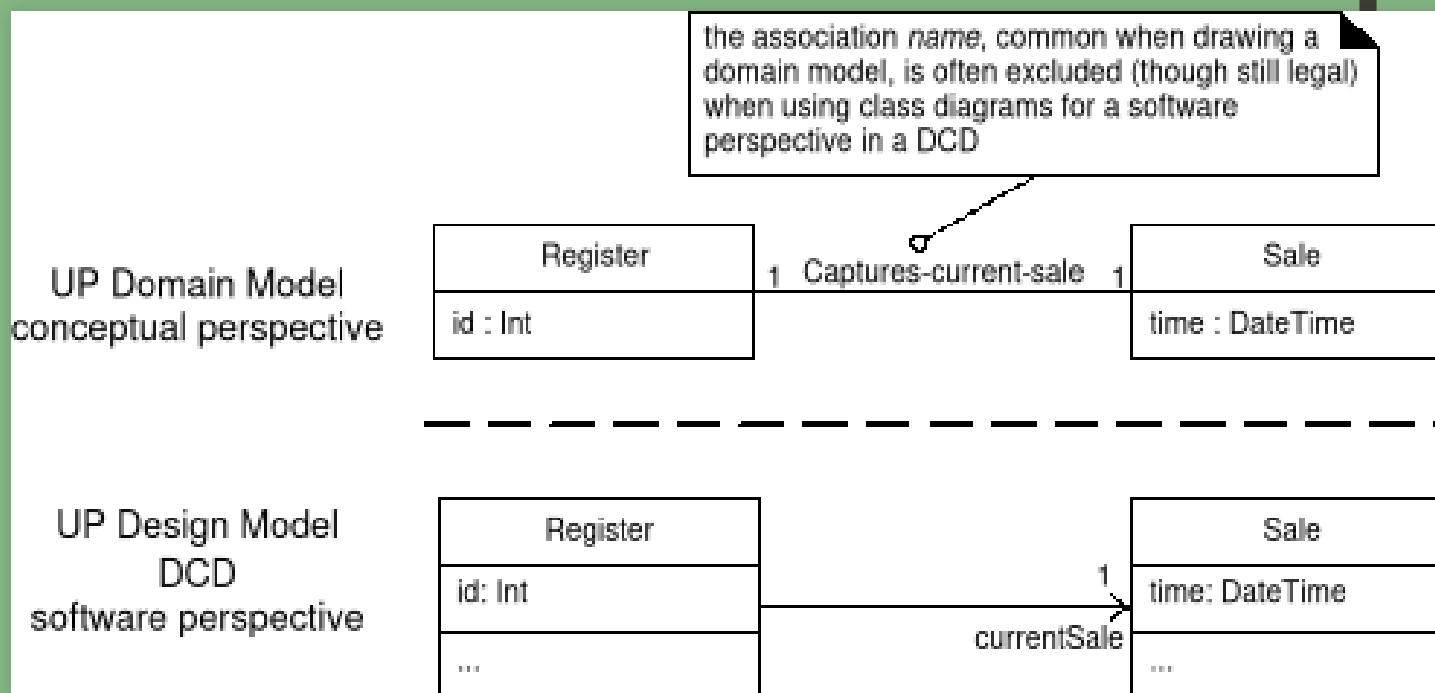


Ben-Peter Ekkelbus  
Project

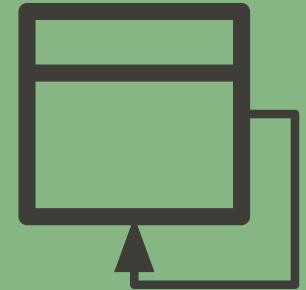
# DÉCALAGE DE PRÉSENTATION



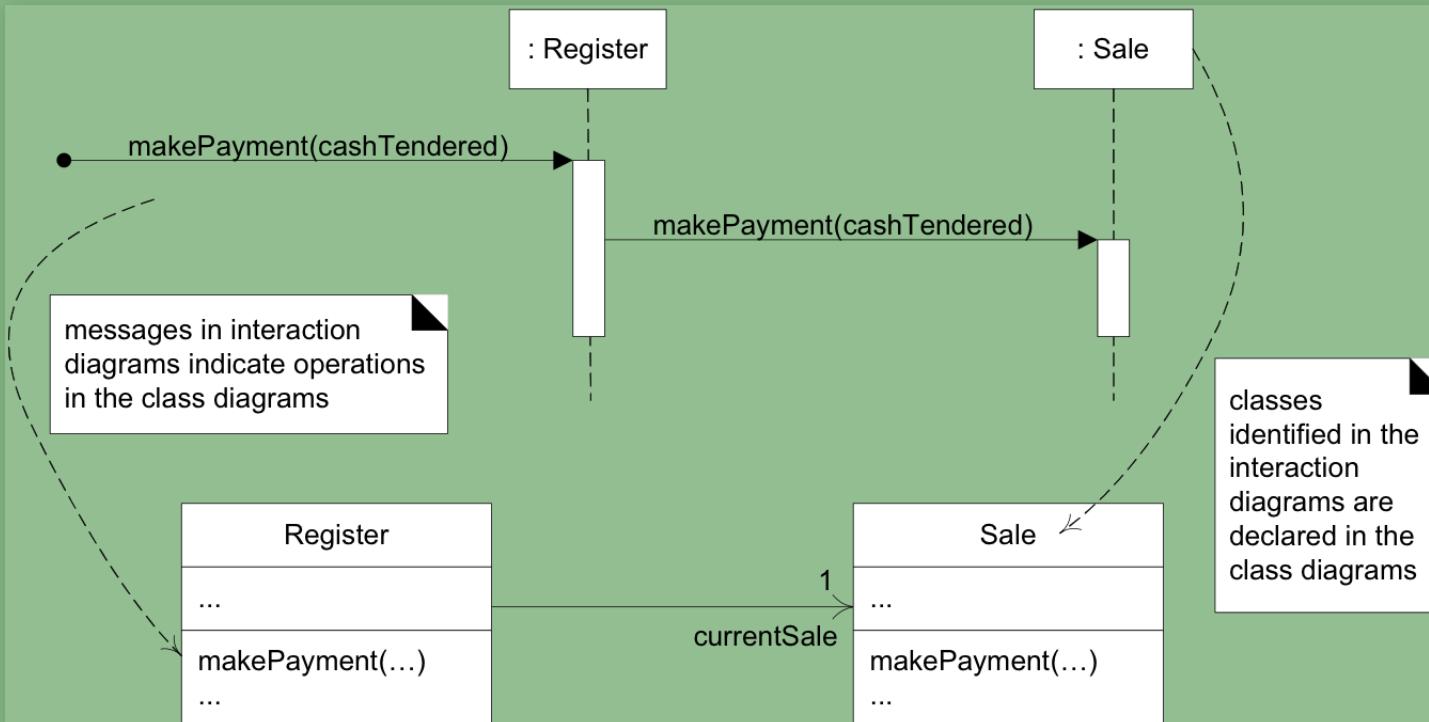
Ben-Peter Ekkelbus  
Project



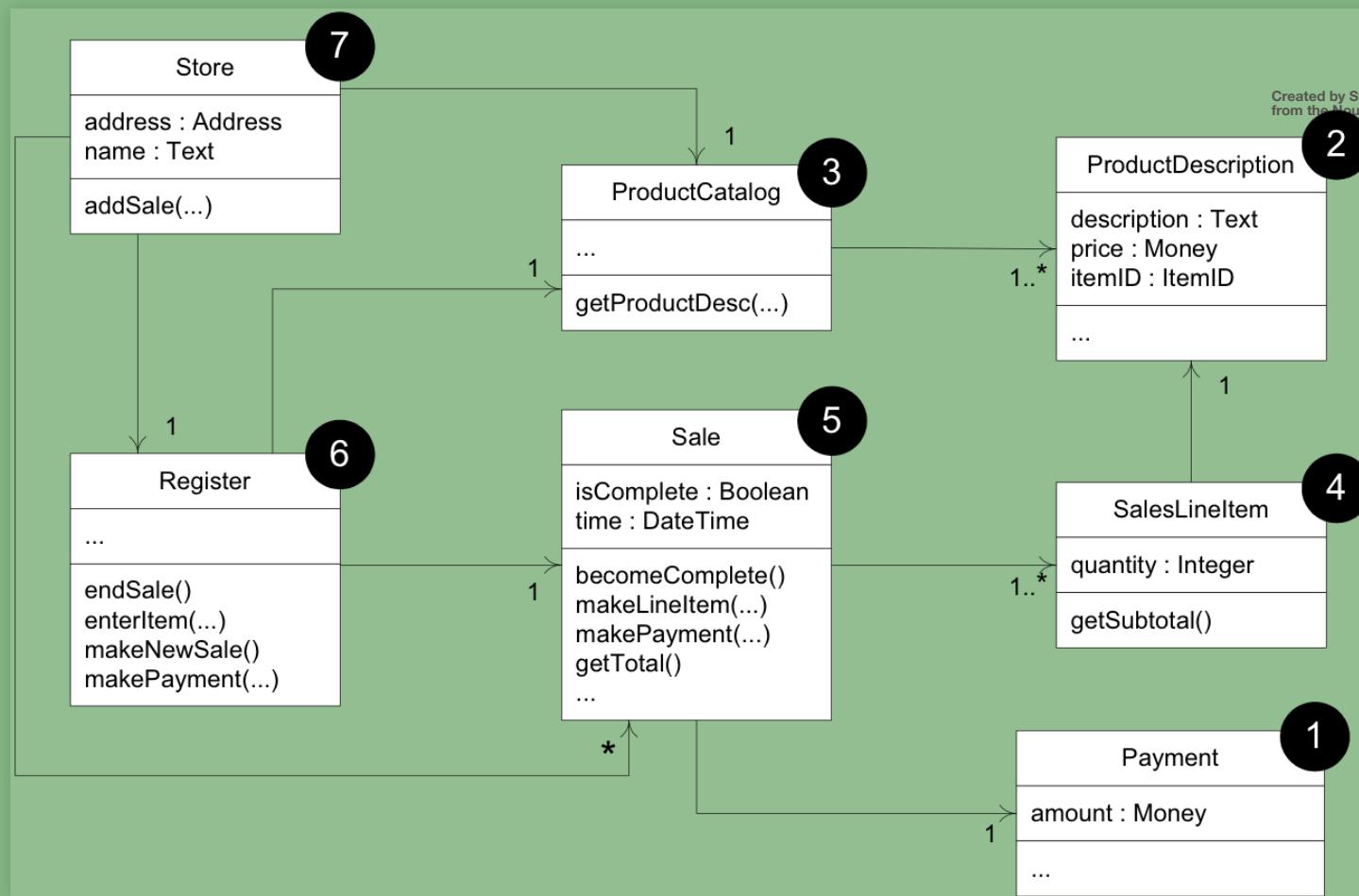
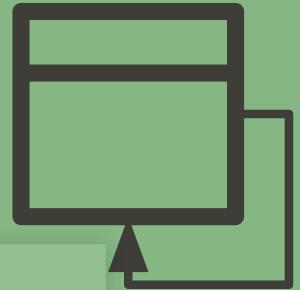
# RELATION DIAGRAMMES D'INTERACTION ET DE CLASSES



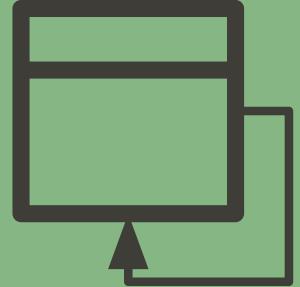
Created by Swen-Peter Ekkebus  
from the Noun Project



# ORDRE D'IMPLEMENTATION



# RÉSUMÉ



- DCC servent comme plan pour le codage
  - Soyez à l'aise avec la transformation
- DCC se construisent au fur et à mesure que l'on réalise la dynamique (à travers les diagrammes d'interaction)

Created by Swen-Peter Ekkebus  
from the Noun Project

# ITÉRATION SUIVANTE

CU02-Le client aimerais pouvoir réserver plusieurs chambres avec des dates différents lors de la même réservation

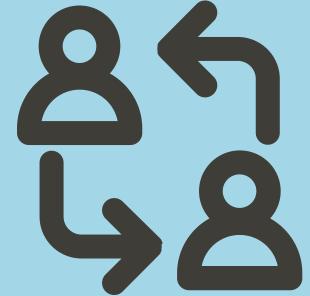
- Qu'est-ce qui change au niveau du DCU
- Qu'est-ce qui change au niveau du CU
- Qu'est-ce qui change au niveau du MDD
- Qu'est-ce qui change au niveau des IU
- Qu'est-ce qui change au niveau du DSS
- Qu'est-ce qui change au niveau des contrats
- Qu'est-ce qui change au niveau des RDCU
- Qu'est-ce qui change au niveau du DCL

Noter une réservation



# SÉANCE #01

## RÉTROACTION: PAGE D'UNE MINUTE



Created by Prithvi  
from the Noun Project

1. Quels sont les deux [trois, quatre, cinq] plus importants [utiles, significatives, surprenantes, dérangeantes] choses que vous avez apprises au cours de cette session?
2. Quelle (s) question (s) reste (s) en tête dans votre esprit?
3. Y a-t-il quelque chose que tu n'as pas compris?

<https://1drv.ms/u/s!An6-F73ulxAOhVyiCB46jTeINVLS>



# To initialise text to speech use the following process

- edit  
/Users/rossypro/vscode/extensions//Users/rossypro/vscode/extensions/reveal-4.0.3/views/initialize.ejs
- in dependencies section add
  - { src: 'libs/reveal.js/3.8.0/plugin/tts\_min/tts\_min\_reveal.js', async: false }
- copy tts\_min directory at /Users/rossypro/vscode/extensions/evilz.vscode-reveal-4.0.3/libs/reveal.js/3.8.0/plugin
- modify tts\_min\_reveal.js to our needs
- Shortcuts
  - “h pour aide, e pour changer de langue, p pour lire, q cancel, t activer/désactiver la voie, r pour enregistrez, z pour récupérer les fichiers audio”

Bonjour je peux lire du texte en français.

Hello my name is Samanta I can read text in english

Bonjour je m'appelle Amélie je suis la voie par défaut de Chrome. Vous avez besoin d'une configuration d'écran de 1920x1048 pour afficher sur les écrans Creston de l'École de Technologie Supérieure