



Génie logiciel

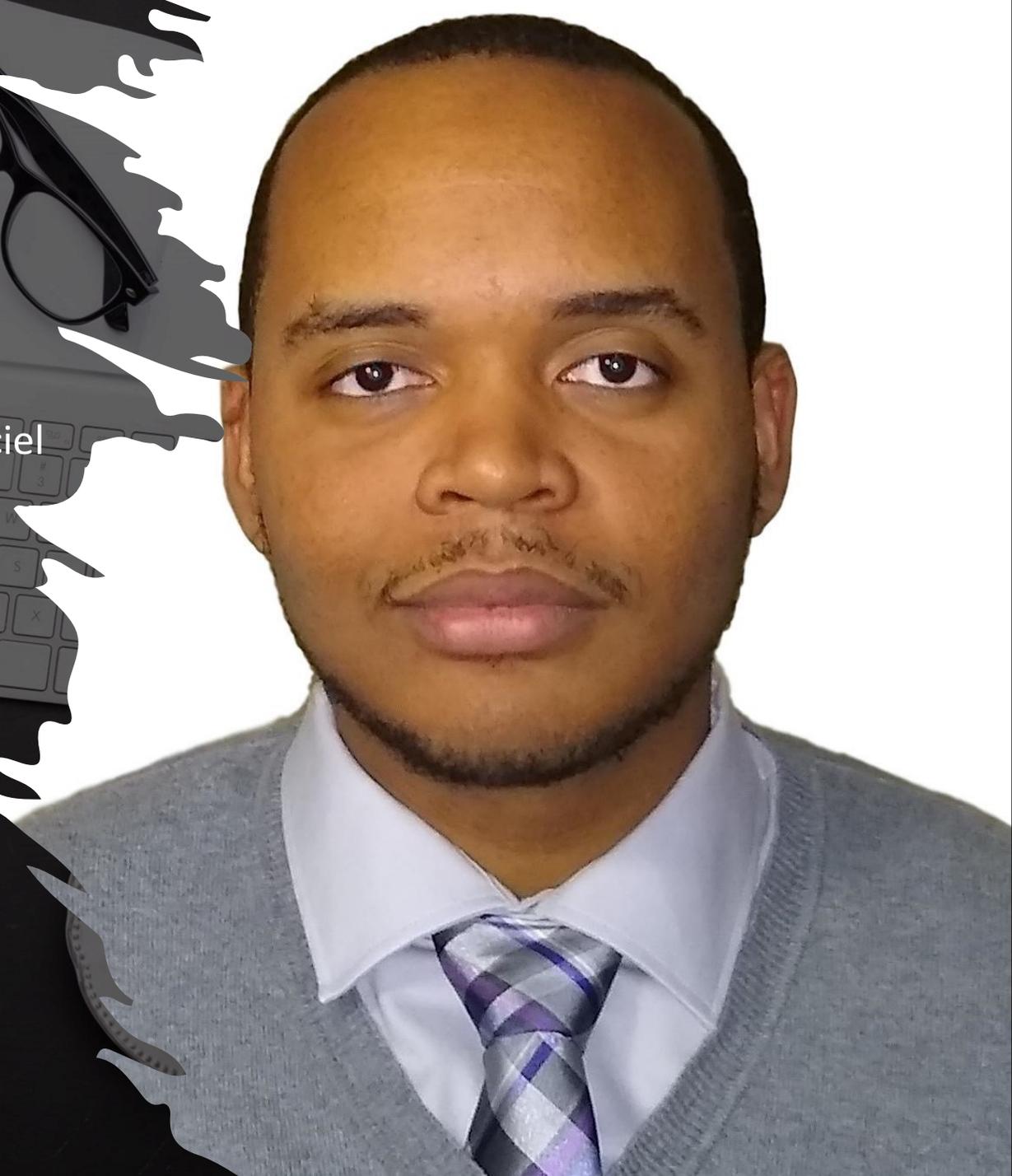
# Introduction

Louis-Edouard LAFONTANT



# Courte bio.

- Étudiant au doctorat en informatique à l'UdeM
- Maîtrise en informatique, spécialisé en génie logiciel
- Membre du **GEODES** – Groupe de recherche du DIRO dédié au génie logiciel
- Intérêts: Interaction humain-machine (HCI), Environnement dynamique, Modélisation, Vulgarisation
- Travaux de recherche
  - Domain-specific modeling
  - Low-code development
  - Projectional editing⇒ **Gentleman** (collab. avec E. SYRIANI)



# Organisation du cours

## StudiUM

Slides du cours et démos  
Quiz et révisions  
Remises des travaux  
Ressources et liens rapides



Annonces et Questions-réponses  
Communication direct avec les enseignants  
Travail en équipe  
Lien: <https://discord.gg/vJQDrbtGEN>

**IFT 2255**  
www

Plan de cours  
Suivi de la semaine (calendrier + résumé)  
Ressources et liens rapides  
Lien: <https://ceduni.github.io/udem-ift2255/aut2022.html>

# Projet

---

Permettre à chacun de faire le suivi de ces déchets domestiques

- 3 remises
  - Remise 1: 10%
  - Remise 2: 15%
  - Remise 3: 25%
- Travail en équipe de 3 ou 4
- Langage de programmation: Java
- Rapport en HTML
- Visite d'un expert (bonus 🙌 )

---

Ceci 🙌 est un vrai projet!





# Génie logiciel

Qu'est-ce que c'est?

**PRESS  
START**



Qu'entend-on par  
genie ou ingénieur?

**ing.** **Ordre  
des ingénieurs  
du Québec**





If you're nothing without the suit, then you shouldn't have it

*L'habit ne fait pas le moine*

Il faut observer la rigueur dans tout ce qu'on fait

**Trust me I'm an  
engineer**



Ce n'est pas  
une question  
de confiance,  
mais de  
**compétences**

# Mauvaise ingénierie à de graves conséquences





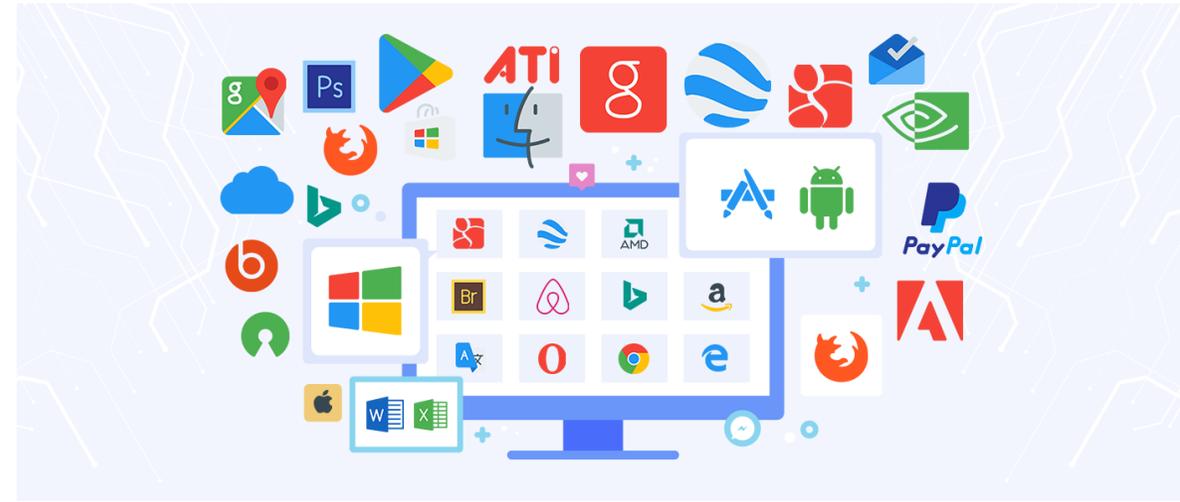
# Génie

## Rigueur et Garantie

- Méthodologie
- Outils
- Observation
- Analyse
- Conception
- Éthique
- ...
- Esprit d'équipe & Communication
- Imagination & Créativité

.....

# Quelle est la nature du logiciel?



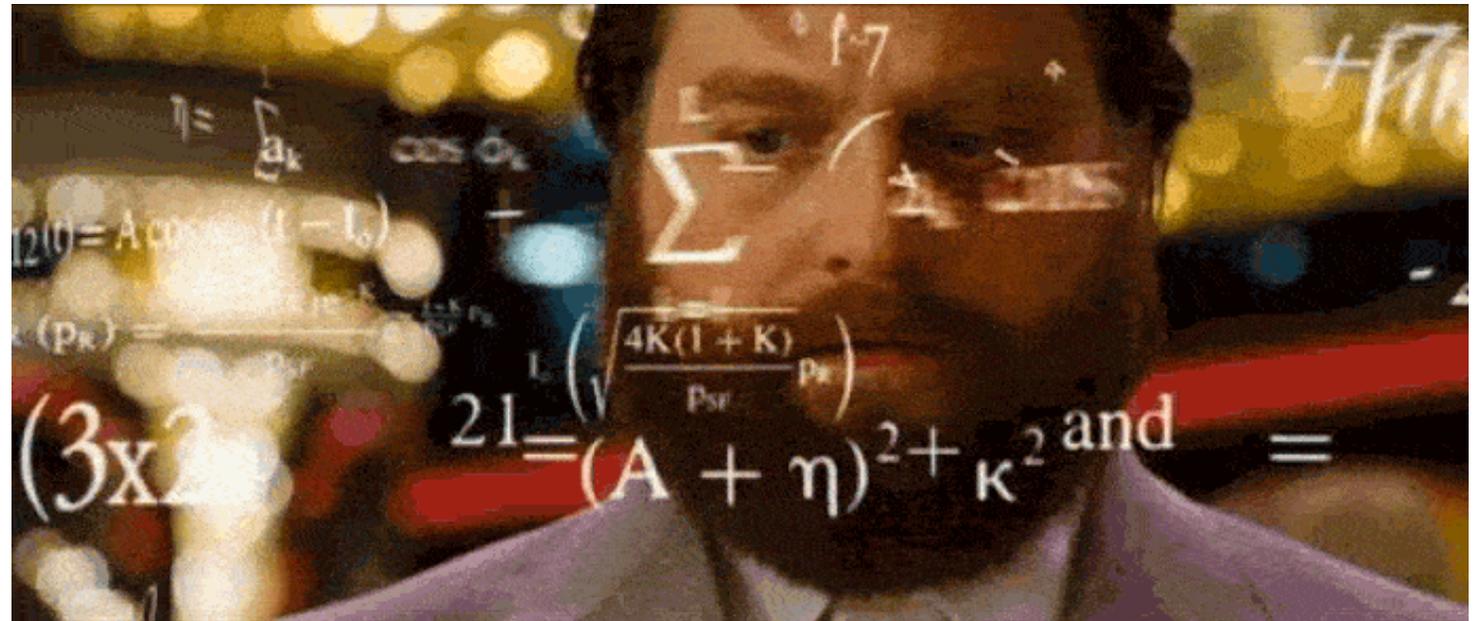
```
Project - /Volumes/Disc/Code/co
  .gitignore
  Collection.php
  js2.js
  js3.js
  php1.php
  php1_edit.php
  scss1.scss
  untitled2.html
  vue_js1.js
  vue_js2.js
  yll.html
  External Libraries
  Scratches and Consoles
  Structure PHP
  MyM
  repo:App\Repositories\ProductRep
  cat_id:int
  __construct()
  index():void
  impdeploy():void

165 public function new_rename()
166 {
167     $bar = new ProgressBar($this->output, 96);
168     $bar->setFormat('debug');
169     $bar->setBarCharacter('<<comment><<comment>');
170     $bar->setBarWidth(50);
171     $bar->start();
172
173     $prod_path = public_path('imgs/products/');
174
175     $products = Product::select('id', 'image')
176         ->where('cat_id', 162)
177         ->where('mirror', 0)
178         ->get();
179
180     foreach ($products as $product) {
181         $img_cuts = explode(delimiter: '.', $product->image);
182         $img_cut = $img_cuts[0].'-'. $img_cuts[1];
183         $min_file = glob(pattern: $prod_path . 'big_min' . $img_cut . '*.jpg');
184         $big_file = glob(pattern: $prod_path . 'big_big' . $img_cut . '*.jpg');
185         rename(current($min_file), rename: $prod_path . 'temp_min' . $product->image);
186         rename(current($big_file), rename: $prod_path . 'temp_big' . $product->image);
187
188         if(!file_exists(filename: public_path('imgs/products/') . $img_cut . $prod->image))
189             DB::table('products')->where('id', $prod->id)->update(['public' => 1]);
190         }else{
191             DB::table('products')->where('id', $prod->id)->update(['public' => 0]);
192         }
193     }
194     $bar->advance();
195 }
```

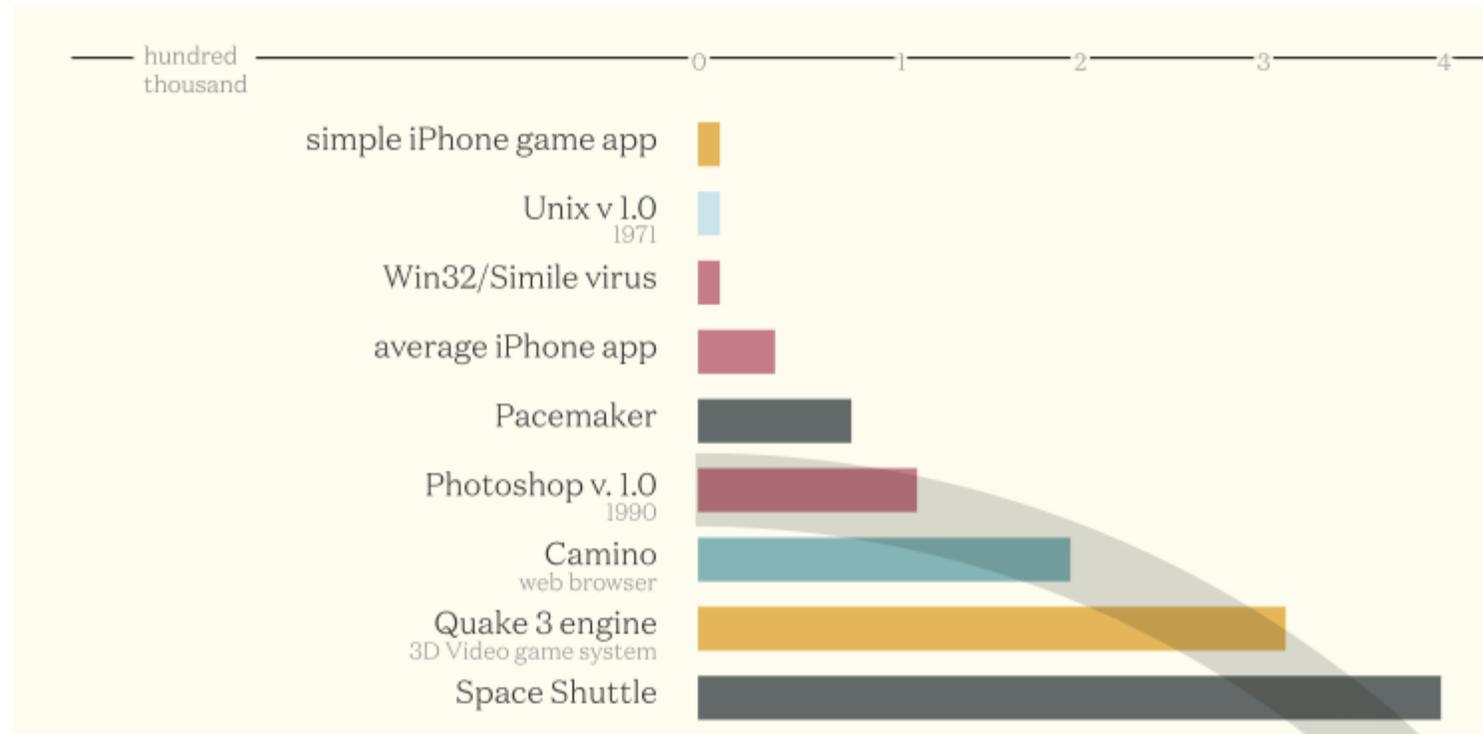


# Complexité

- Les programmes sont parmi les systèmes les plus complexes jamais créés
- Notre mémoire à court terme peut accommoder  $\pm 7$  choses (Miller, 1956)



# Programmes en nombre de ligne de codes



<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>

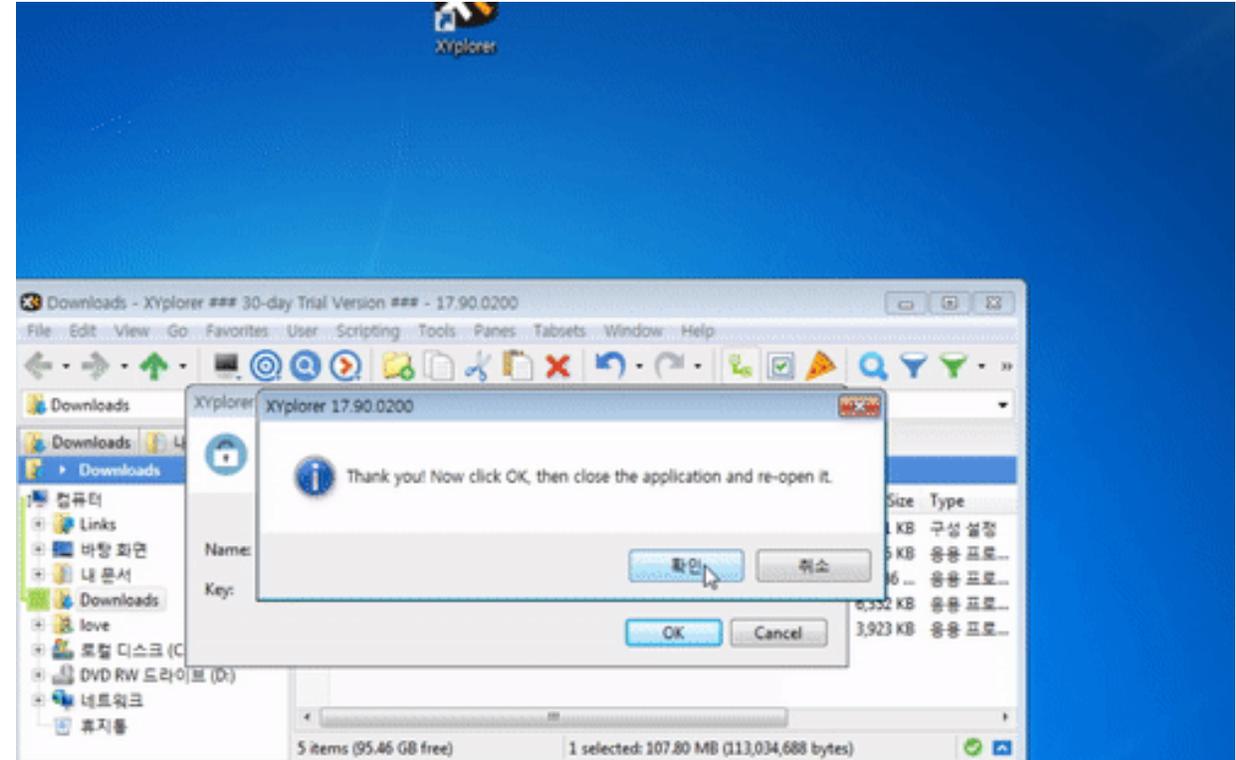
# Invisible

- Pouvez-vous toucher une classe ou une fonction?
- Diffère des lois de la physique et des mathématiques (continues)
- Pas moyen de représenter un produit au complet
- Nos sens ne peuvent pas être facilement utilisés pour comprendre le logiciel



# Discontinuité

- L'humain comprend facilement les systèmes linéaires ou semi-linéaires
- Les logiciels sont discontinus: petits changements en entrée résultent en un énorme changement en sortie
- Le feedback est insuffisant pour permettre à l'utilisateur d'anticiper certains comportements



# Versatilité

- Logiciels évoluent et changent constamment
  - Changement dans les besoins
- Ce qu'on savait hier peut être obsolète aujourd'hui
  - Nouvelle/meilleure techno émerge continuellement (loi de Moore)
- Le logiciel est facile à modifier
- Mais la défi est de le modifier **correctement**

# Logiciel

## Complexité accidentelle

- Due aux technologies utilisées
- Imprévus de l'environnement
- Problèmes transitoires

## Complexité essentielle

- Complexité
- Versatilité
- Invisibilité
- Discontinuité

# Origines du Génie logiciel

[Rapport conférence 1968](#)

## SOFTWARE ENGINEERING

Report on a conference sponsored by the  
NATO SCIENCE COMMITTEE

Garmisch, Germany, 7th to 11th October 1968

# Origines du génie logiciel

- Jusqu'au milieu des années 1960, logiciels construits à l'improviste
- Programmes codés par des experts de divers métiers
  - Ingénieurs matériels, mathématiciens
  - Le temps d'utilisation d'un ordinateur coûtait plus cher (600\$/h) que le salaire de ses opérateurs (2\$/h)
- Programmes devenaient de plus en plus complexes avec de nouvelles techno, des besoins qui évoluent et la diversité des programmeurs
- Il n'y avait pas de méthodologie pour construire un logiciel et le **changer**
  - Coder puis corriger (*code-and-fix programming*)

# Naissance du génie logiciel

- En 1968, la conférence de l'OTAN se discute de la création d'un nouveau domaine: le Génie logiciel
  - ⇒ Considérer le Génie logiciel comme une discipline à part entière
  - ⇒ Suivre les mêmes méthodologies que les disciplines de génie traditionnelles
- **Participants majeurs:** Dijkstra, Naur, Bauer, Perlis, Gries, McIlroy, Randell
  - ⇒ Ils proposent des recommandations sur comment développer du logiciel



# Activités de l'ingénieur logiciel



**Décrire:** besoins, spécification de conception, documentation



**Implémenter:** conception, programmation



**Évaluer:** test, vérification, validation, révision



**Gérer:** planification, échelonnage, communication



**Faire fonctionner:** déploiement, installation, maintenance