



Ingénierie Logicielle

Introduction à l'Ingénierie Logicielle

Prof. Ousmane SALL, Maître de Conférences CAMES en Informatique



Objectifs de la séquence

- A l'issue de cette séquence vous serez capable de :
 - Définir ce qu'est l'Ingénierie logicielle ou génie logiciel, en comprendre les enjeux.
 - Décrire les qualités d'un logiciel et les principes de base qui permettent de les réaliser.



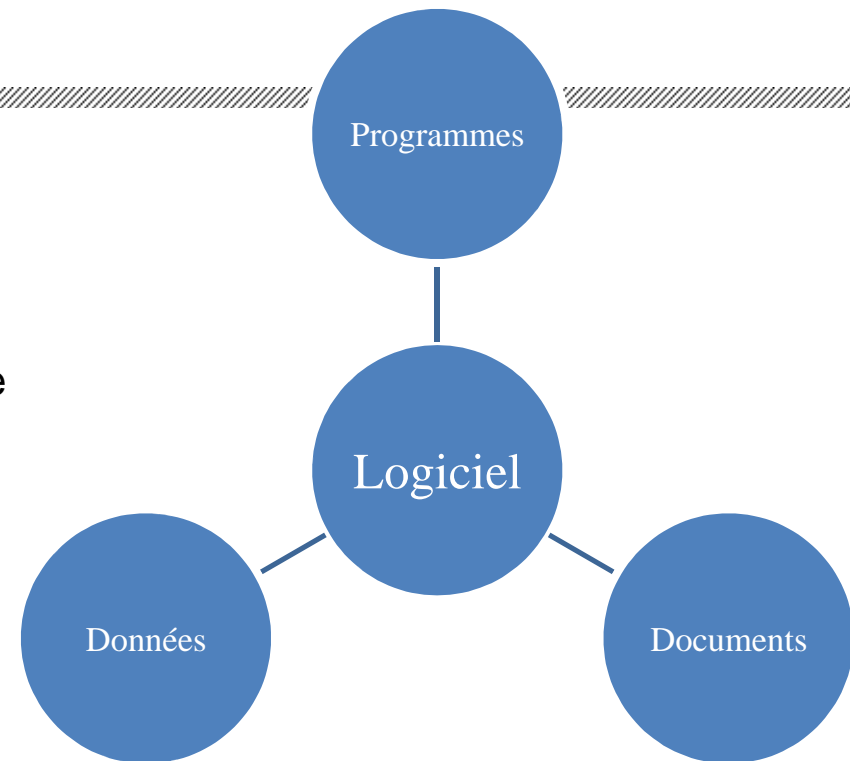
Les logiciels et la société



Introduction à l'Ingénierie logicielle: **Le Logiciel**

Qu'est-ce qu'un logiciel ?

- **Un logiciel est:**
 - Du **code exécutable**
 - Les **données** associées au programme
 - Des **documents**: besoins des usagers, design, guide d'utilisateur, guide de programmation, etc.
- **Les logiciels jouent un rôle clé:**
 - Produisent, gèrent et présentent de l'information



- Le logiciel est un produit que les ingénieurs en logiciel conçoivent et construisent
- Il comprend des programmes, des documents et des données

Question ?

En quoi l'Ingénierie d'un logiciel est-elle différente de celle des autres produits industriels classiques ?



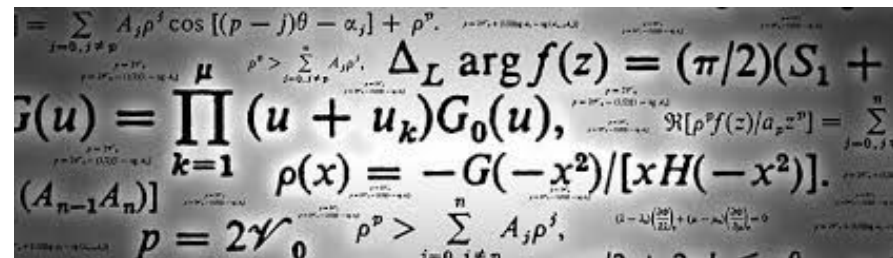
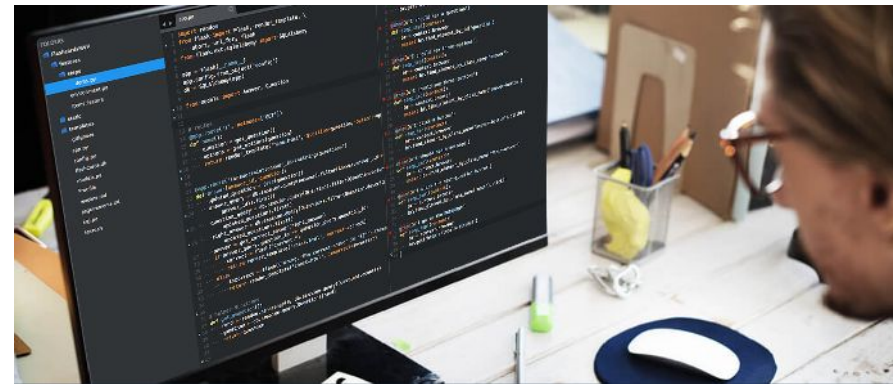
Le logiciel est différent parce qu'il...

- Est invisible/intangible.
- Ne vieillit pas.
- Ne s'use pas, il n'est pas détérioré par le test.
- Est fabriqué artisanalement.
- Est très (trop ?) facile à modifier.
- Coûte très (trop ?) cher.



Le logiciel est différent parce que...

- Nous ne pouvons **pas voir** le logiciel en cours de développement pour déterminer ses progrès.
- Mais on peut voir que le pont est construit pour déterminer ses progrès.
- Nous ne pouvons **pas prouver** que les logiciels n'échoueront pas.
- Mais nous pouvons prouver qu'un pont ne tombera pas.





BREAKING NEWS
the NOW SELF-DRIVING WAYMO INVOLVED IN CRASH

abc 15
ARIZONA
4:01 91°



TEMPE

abc 15
ARIZONA

DEADLY CRASH WITH SELF-DRIVING UBER



GOOGLE SELF-DRIVING CAR GETS INTO AN ACCIDENT INVOLVING INJURIES

Étant donné que.....

- Les échecs logiciels ont des effets dévastateurs
- Le logiciel est différent des autres produits
- Il n'est pas facile de gérer et de développer des logiciels.

Pouvons-nous faire quelque chose?



Oui...

La solution est

Ingénierie Logicielle ou Génie Logiciel



History of software engineering

The notion of 'software engineering' was first proposed in 1968 at a conference held to discuss what was then called the 'software crisis' (Naur and Randell, 1969). It became clear that individual approaches to program development did not scale up to large and complex software systems. These were unreliable, cost more than expected, and were delivered late.

Throughout the 1970s and 1980s, a variety of new software engineering techniques and methods were developed, such as structured programming, information hiding and object-oriented development. Tools and standard notations were developed and are now extensively used.

<http://www.SoftwareEngineering-9.com/Web/History/>

<https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/Web/History/>

Introduction à l'Ingénierie logicielle: Génie Logiciel (Ingénierie Logicielle)

Génie Logiciel

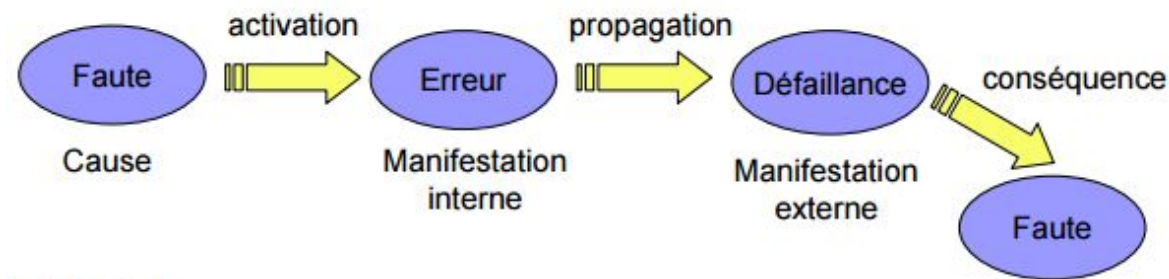
- **Definition:** Le **génie logiciel**(Software Engineering) est un processus méthodologique pour construire un logiciel dans les **délais** prévus, un **budget** maîtrisé, des **performances acceptables** avec un **fonctionnement correct**.



- Les économies de tous les pays développés dépendent des logiciels.
- De plus en plus de systèmes sont contrôlés par logiciel.
- Le génie logiciel s'intéresse aux théories, méthodes et outils de développement de logiciels professionnels.
- Les dépenses en génie logiciel représentent une fraction importante du PNB des pays développés.

Les préoccupations du génie logiciel

- Prévenir, tolérer et éliminer les fautes:
 - **Défaillance (panne)**: Survient quand le comportement d'un système viole sa spécification de service (i.e. violation du contrat entre le fournisseur et l'utilisateur). Situation symptomatique ou la manifestation externe d'une erreur.
 - **Erreur**: Partie de l'état d'un système susceptible de causer une défaillance.
 - **Faute**: Toute cause adjugée ou hypothétique (événement, action, circonstance) pouvant provoquer une erreur.

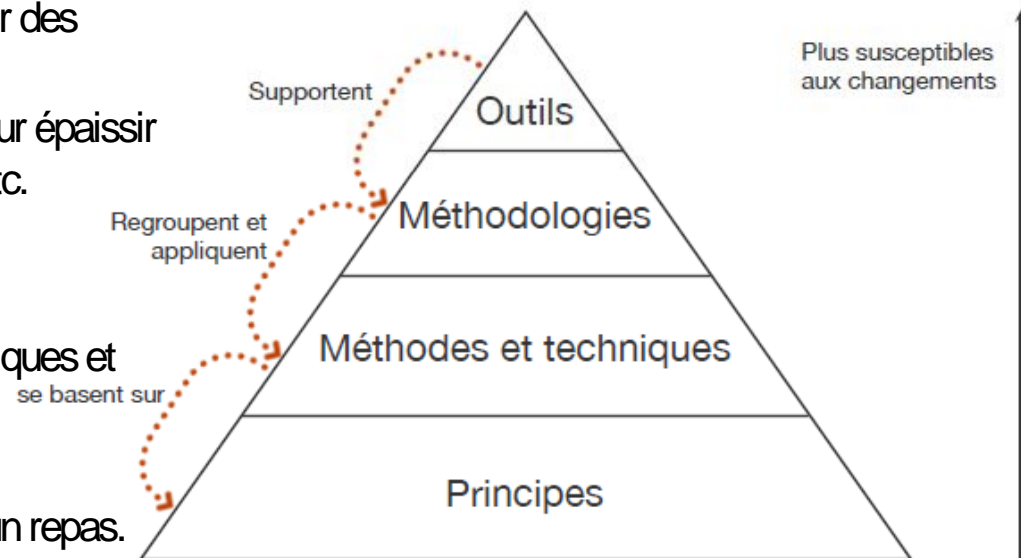


Principes du génie logiciel

- Rigueur et formalité
- Séparation des préoccupations
- Modularité
- Abstraction
- Anticipation du changement
- Généralité
- Construction incrémentale

Génie logiciel : Méthodes, techniques et outils

- Développer un logiciel c'est comme cuisiner.
 - **Principes** = Règles d'hygiène, fraîcheur des aliments.
 - **Techniques** = techniques culinaires pour épaissir une sauce, fouetter une mayonnaise, etc.
 - **Méthodes** = les recettes
 - **Outils** = casseroles, bols, mixette, etc.
 - **Méthodologie** = livre de cuisine (techniques et méthodes)
 - **Paradigme** = style de cuisine
 - **Processus** = étapes de préparation d'un repas.
 - Prochaine étape ? Étape finie ?



Génie logiciel : Méthodes, techniques et outils

Outils de production

- Organisation: Make, ANT,...
- Documentation: Javadoc et Doxygen...
- Test: Junit et Nounit...
- Versions: Cvs,...
- Environnement de développement : Eclipse, NetBeans,...
- Environnements de développement graphiques qui facilitent la création rapide de logiciels
 - Ce sont des programmes utilisés pour prendre en charge des activités de processus logiciel telles que la modélisation, le débogage et les tests de systèmes.
 - Certains outils CASE(Atelier de Génie Logiciel) incluent la génération automatique de code à partir de diagrammes.
 - Upper-Case: outils utilisés pour prendre en charge les premiers processus de conception et d'exigences
 - Lower-CASE: outils pour prendre en charge des activités ultérieures telles que la programmation, le débogage et les tests.

Défis du génie logiciel

- Assurer la **qualité** des logiciels produits
- Minimiser les **coûts** de développement tout en répondant aux exigences croissantes
- Réduire les **temps** de développement
- Instituer l'usage des **nouvelles technologies** (méthodes et outils) du génie logiciel
- Arriver à simultanément:
 - Assurer la maintenance du nombre croissant de logiciels (vieillissants!)
 - Conserver un rythme de production logicielle qui puisse répondre à la demande

Coût

- Les coûts des logiciels dominent souvent les coûts du système. Les coûts des logiciels sur un PC sont souvent supérieurs aux coûts matériels.
- Pour les systèmes à longue durée de vie, les coûts de maintenance peuvent représenter plusieurs fois les coûts de développement.
- Les logiciels coûtent plus cher à entretenir qu'à développer.
- Le génie logiciel est concerné par le développement logiciel rentable.

Environ 60% des coûts logiciels sont liés au développement frais; 40% sont des coûts de test. Pour les logiciels personnalisés, les coûts d'évolution dépassent souvent les coûts de développement. *Ian Sommerville*

Introduction à l'Ingénierie logicielle: **Qualités d'un bon logiciel...**

Qualités d'un bon logiciel

- **Correct:** programme fiable sans “bogue”
 - Le programme est conforme à sa spécification
 - Il est testé sur plusieurs cas, normaux et extrêmes
- **Convivial:** facile à utiliser
 - Interface usager intuitive, cohérente et esthétique
- **Performant:** rapide et économe en mémoire
 - Pas besoin d'un périphérique(ordinateur,...) dispendieux
 - Logiciel “réactif” (contrôle, jeux, ...)
 - “Scalable”

- **Facile à maintenir: modifiable et extensible**

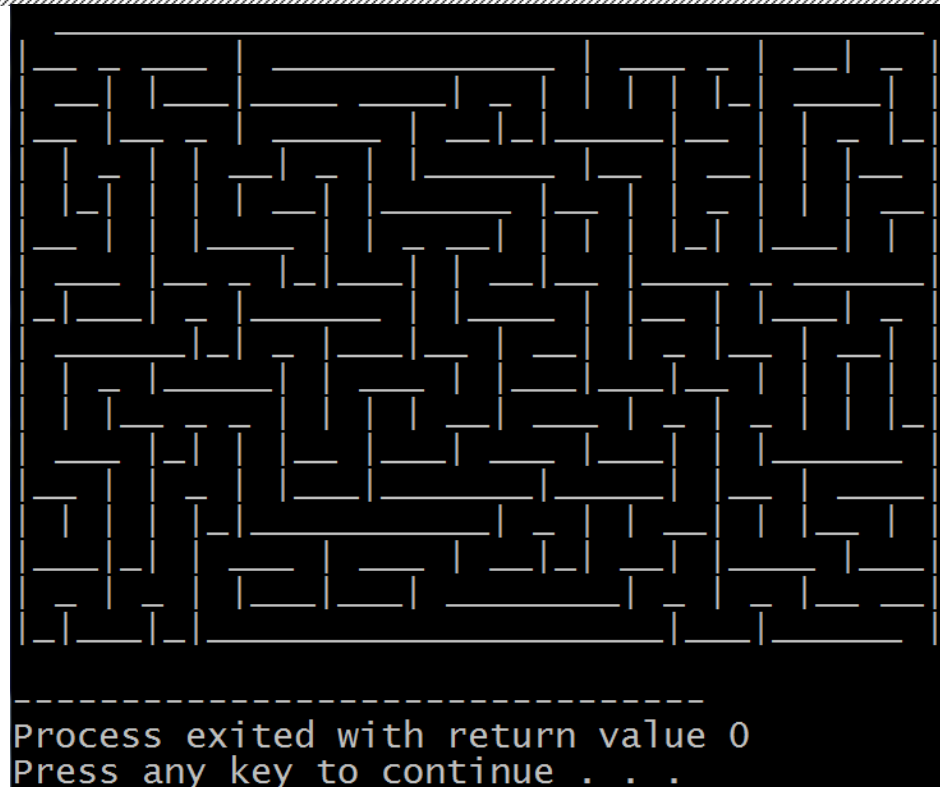
- Chaque logiciel a un cycle de vie
 1. *Analyse et conception:* analyse des besoins, spécification
 2. *Implémentation:* développement d'algorithmes, codage
 3. *Maintenance:* réparer bogues, extension
- Cycle de vie: quelques minutes à des dizaines d'années
- Concepts de: portabilité, réutilisabilité, modularité, tests de régression, lisibilité, standards et styles de programmation



Que fait ce programme ?

```
1 #define P(X)j=write(1,X,1)
2 #define C 21
3 int M[5000]={2},*u=M,N[5000],R=17,a[4],l[ ]=
4 {0,-1,C-1,-1},m[ ]={1,-
5 C,-1,C},*b=N,*d=N,c,e,f,g,i,j,k,s;main()
6 {for(M[i=C*R-1]=24;f|d>= b;)
7 {c=M[g=i];i=e;for(s=f=0;s<4;s++)if((k=m[s]
8 +g)>=0&&k<C*R&&l[s]!=k%C&&(!M[k]||!j&&c>= 16!=
9 M[k]>=16))a[f++]=s;rand();if(f){f=M[e=m [s=a[rand()
10 %f]]+g];j=j<f?f:j;f+=c&-16*!j;M[g]=c|
11 1<<s;M[*d++=e]=f|1<<(s+2)%4;}else e=d>
12 b++?b[-1]:e;}P(" ");P("");for(s=C;--s;P("_" ))
13 P("_");for(;P("\n"),R--;P("|"))for(e=C;e-- ;
14 P("_ " + (*u++/8)%2))P("|_" + (*u/4)%2);}
```

Ceci, bien sûr !



Maintenabilité

- Définition
 - ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données
- Sous-caractéristiques:
 - **Facilité d'analyse** : effort nécessaire pour diagnostiquer les déficiences et causes de défaillance ou pour identifier les parties à modifier
 - **Facilité de modification** : effort nécessaire pour modifier, remédier aux défauts ou changer d'environnement
 - **Stabilité** : risque des effets inattendus des modifications
 - **Facilité de test** : effort nécessaire pour valider le logiciel modifié

Les changements exigent que les méthodes de développement tiennent compte du changement et s'y adaptent efficacement. Les demandes de changement peuvent perturber un projet et, si elles ne sont pas traitées correctement, peuvent représenter jusqu'à 30 à 40% des coûts de développement.

- **Systèmes d'information et de commerce électronique**

- Gestion d'information: création/destruction, consultation et mise à jour de données. Ex: systèmes bancaires, catalogue informatisé (biblio)
- Caractéristiques importantes
 - Intégrité des données
 - Sécurité
 - Disponibilité des données
 - Performance des transactions
 - Convivialité



- **Systemes temps réel**

- Doit répondre aux événements dans un laps de temps limité.
- Temps de réponse fait partie de la fonctionnalité du logiciel
- Ex.: système d'injection d'un moteur à combustion, système de guidage d'une sonde
- Caractéristiques importantes
 - Correction, fiabilité, robustesse
 - Interopérabilité, convivialité



Exigences de qualité dans différents domaines

- **Systèmes distribués**

- Ordinateurs indépendants connectés par un réseau de communication.
- Aspects à considérer
 - Distribution: données ou contrôle?
 - Tolérance aux fautes
- Caractéristiques importantes
 - Fiabilité, robustesse
 - Performance
 - Interopérabilité



Exigences de qualité dans différents domaines

- **Systèmes embarqués**

- Système intégré à un dispositif, une machine ou un autre système pour le piloter.
- Exemples: système de freinage automobile, tableau de commande d'un micro-ondes, etc.
- Aspect à considérer
 - Interface avec autre système (non humain).
- Caractéristiques importantes
 - Interopérabilité
 - Réutilisation



Introduction à l'Ingénierie Logicielle:

Ethique du génie logiciel

Ethique du génie logiciel

- L'ingénierie logicielle implique *des responsabilités plus larges que la simple application de compétences techniques*.
- Les ingénieurs en logiciel doivent se *comporter* de manière *honnête* et *éthiquement responsable* s'ils veulent être respectés en tant que professionnels.
- Un comportement éthique ne se limite pas au respect de la loi, mais implique le respect d'un ensemble de principes moralement corrects.
- **Tel que recommandé par le groupe de travail de l'IEEE-CS/CAM sur l'éthique et la déontologie en génie logiciel et approuvé conjointement par l'ACM et le IEEE-CS en tant que norme d'enseignement et de pratique du génie logiciel.** <http://seeri.etsu.edu/TheSECode.htm>

Ethique du génie logiciel: Principes

- Les ingénieurs logiciels doivent s'engager à faire de l'analyse, de la spécification, de la conception, du développement, du test et de la maintenance une profession utile et respectée. En accord avec leur engagement à la santé, la sécurité et le bien du public, les ingénieurs logiciels doivent adhérer aux principes suivants:
 - **Le Public.** Les ingénieurs logiciels doivent agir dans l'intérêt public en tout temps.
 - **Le client et l'employeur.** Les ingénieurs logiciels doivent agir d'une manière qui sert le mieux possible les intérêts de leurs clients et de leur employeur, toujours en fonction de l'intérêt public.
 - **Le produit.** Les ingénieurs logiciels doivent s'assurer que leurs produits et les modifications connexes sont conformes aux normes professionnelles les plus élevées possible.

https://ethics.acm.org/wp-content/uploads/2016/07/seeri.french.code_.one_.column.pdf?db00a4

Ethique du génie logiciel: Principes

- En accord avec leur engagement à la santé, la sécurité et le bien du public, les ingénieurs logiciels doivent adhérer aux principes suivants:
 - **Le jugement.** Les ingénieurs logiciels doivent maintenir leur intégrité et leur indépendance dans leur jugement professionnel.
 - **La gestion.** Les gestionnaires et les responsables de génie logiciel doivent souscrire à une approche éthique de la gestion du développement et de la maintenance des logiciels et s'employer à en faire la promotion.
 - **La profession.** Les ingénieurs logiciels doivent s'assurer de l'intégrité et la réputation de la profession en tenant compte de l'intérêt public.
 - **Les collègues.** Les ingénieurs logiciels doivent être justes et appuyer leurs collègues.
 - **Soi-même.** Les ingénieurs logiciels doivent être en situation d'apprentissage continu et promouvoir une approche éthique à la pratique de leur profession.

https://ethics.acm.org/wp-content/uploads/2016/07/seeri.french.code_.one_.column.pdf?db00a4

Introduction à l'Ingénierie Logicielle: **A Retenir...**

A retenir...

- Le génie logiciel est une discipline d'ingénierie qui concerne tous les aspects de la production du logiciel.
- Le logiciel n'est pas simplement un programme ou des programmes, il inclut également une documentation. Les attributs essentiels d'un logiciel de qualité sont la maintenabilité, la fiabilité, la sécurité, l'efficacité et l'acceptabilité.
- Le processus logiciel inclut toutes les activités impliquées dans le développement logiciel. **Les activités de haut niveau comme la** spécification, développement, validation et évolution font partie intégrante de tous processus de développement logiciel.

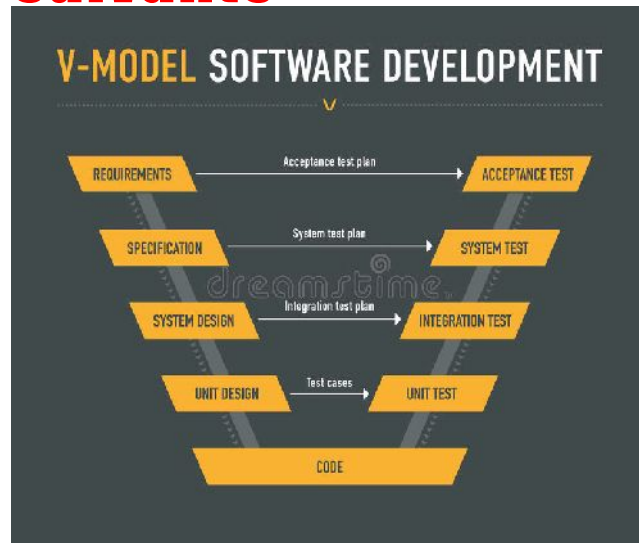
A retenir...

- Les notions fondamentales du génie logiciel s'appliquent universellement à tous les types de développement logiciels. Ces principes fondamentaux incluent les processus de développement logiciels, la fiabilité, la sécurité, exigences et réutilisation.
- Il existe de nombreux types de systèmes et chacun nécessite une approche, des outils et techniques de génie logiciel appropriés pour leur développement. Il y a peu, voire pas du tout, de conception spécifique et techniques de mise en œuvre applicables à tous types de systèmes.

A retenir...

- Les idées fondamentales du génie logiciel s'appliquent à tous les types de systèmes logiciels. Ces principes fondamentaux incluent le management des processus logiciels, la fiabilité et la sécurité des logiciels, ingénierie des exigences et réutilisation des logiciels.
- Les ingénieurs en logiciel ont des responsabilités envers la profession d'ingénieur et la société. Ils ne devraient pas seulement se préoccuper des problèmes techniques.
- Les entreprises publient des codes de conduite énonçant les normes de comportement attendues de leurs membres.

Processus de développement logiciel: Séquence suivante



<https://fr.dreamstime.com/illustration-stock-0403949verloppemen-0403949verloppemen-0403949verloppemen-image74343533>

