Cours GL
exposé 1: Approches et Modèles

2LMD

2016-2017

Laid Kahloul

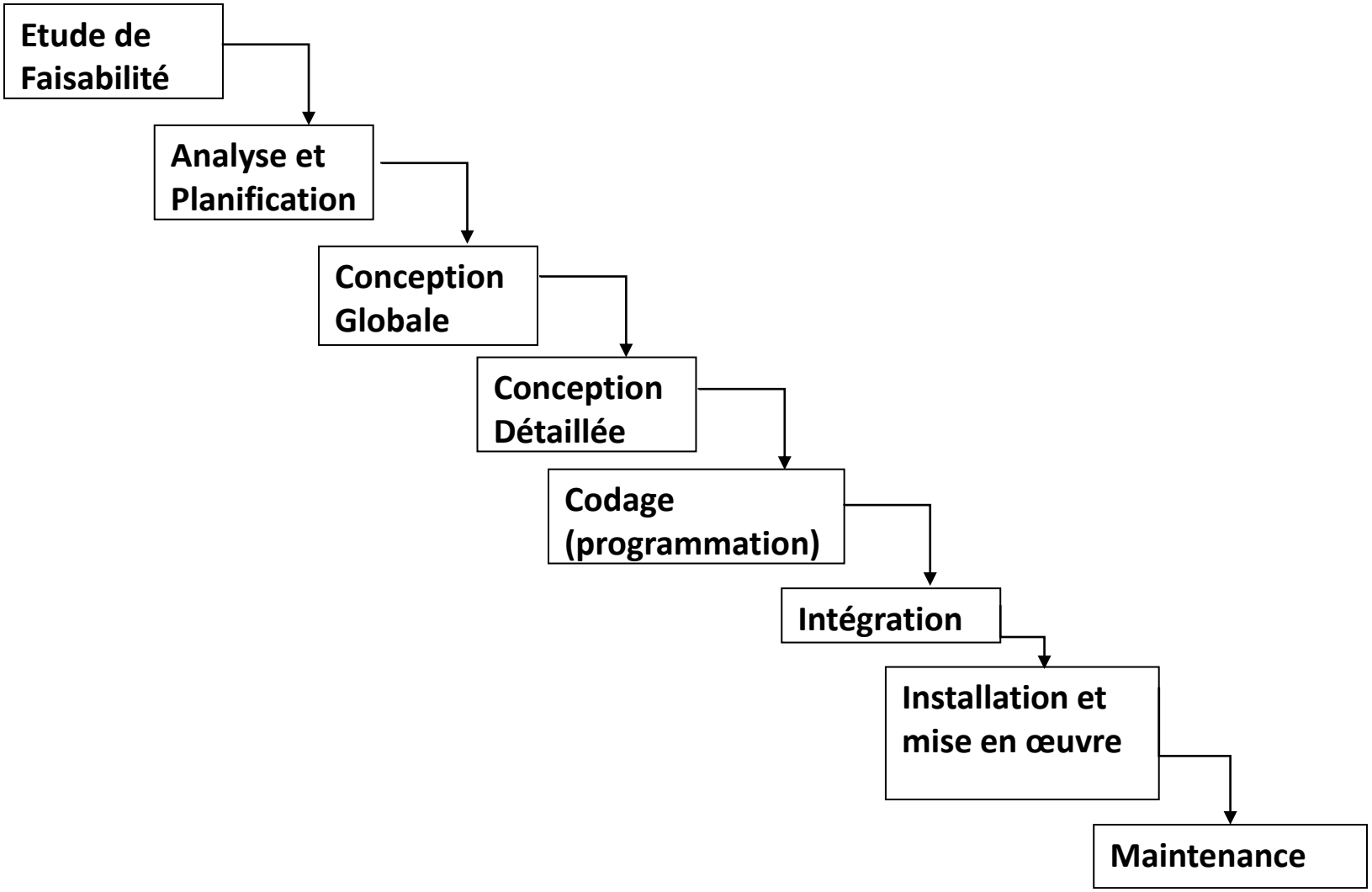
Plan du cours

- **Approches de développement**: Cascade, Programmation exploratoire, Prototypage, Transformation formelles, Réutilisabilité

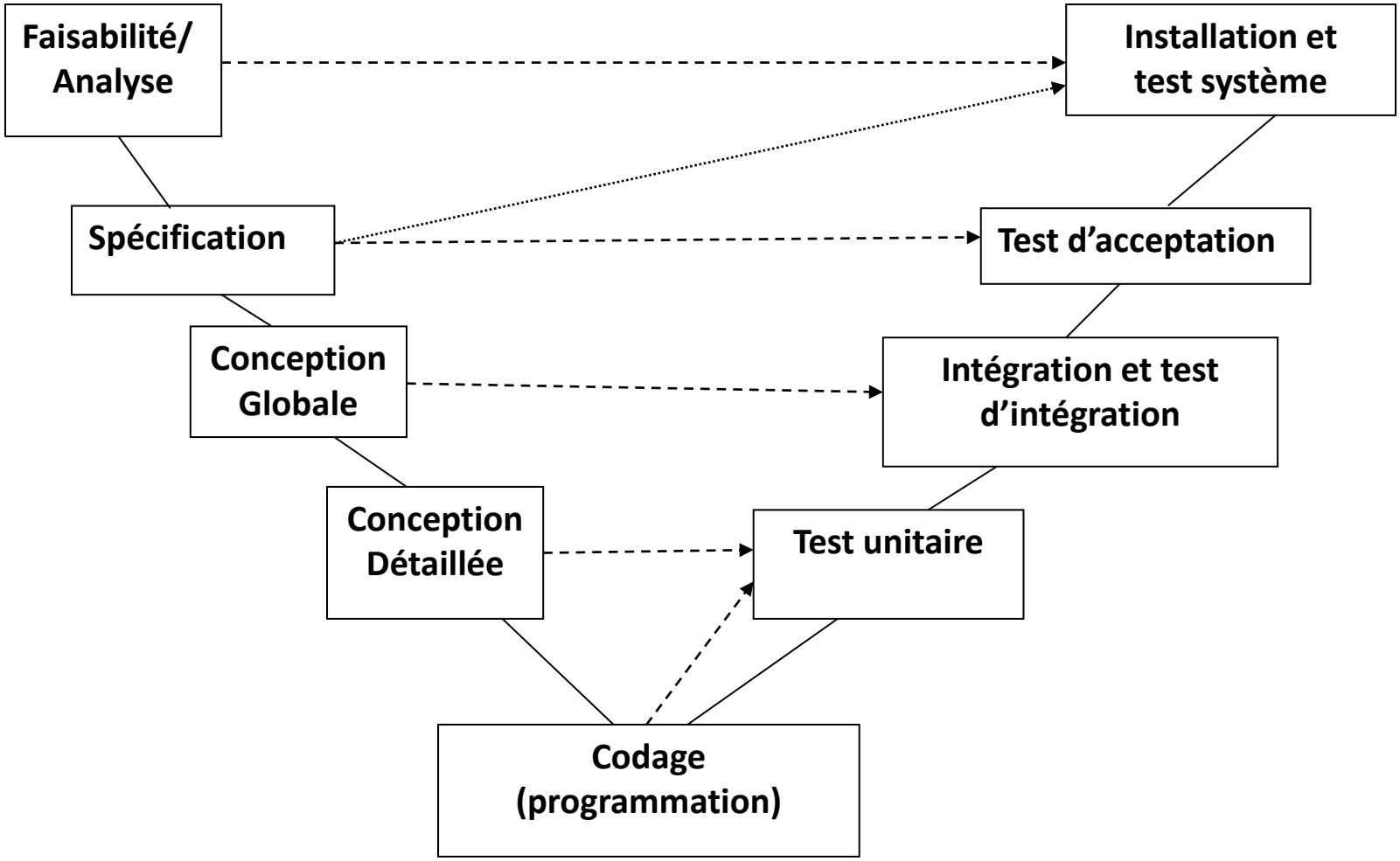
- **Modèles de gestion de projets logiciels**: modèle en V, modèle des incréments;

- GL dans la pratique.

Rappel (2):
Modèle de la cascade



Rappel (3)
Le modèle en V



Des chiffres sur l'usage de la cascade

Systèmes\ ctivité	Analyse et Conception	codage	test
Système de contrôle	16%	20%	64%
Aéronautique	34%	20%	46%
Système d'exploitation	33%	17%	50%
Système de Gestion	44%	28%	28%
Système de calcul scientifique	44%	26%	30%

Coûts des différentes activités pour différents types de systèmes

Approche de la **Programmation exploratoire**

Approches de développement (1)
la programmation exploratoire(1)

- **Objectifs**: présenter rapidement une version opérationnelle au client.
- **Étapes**:

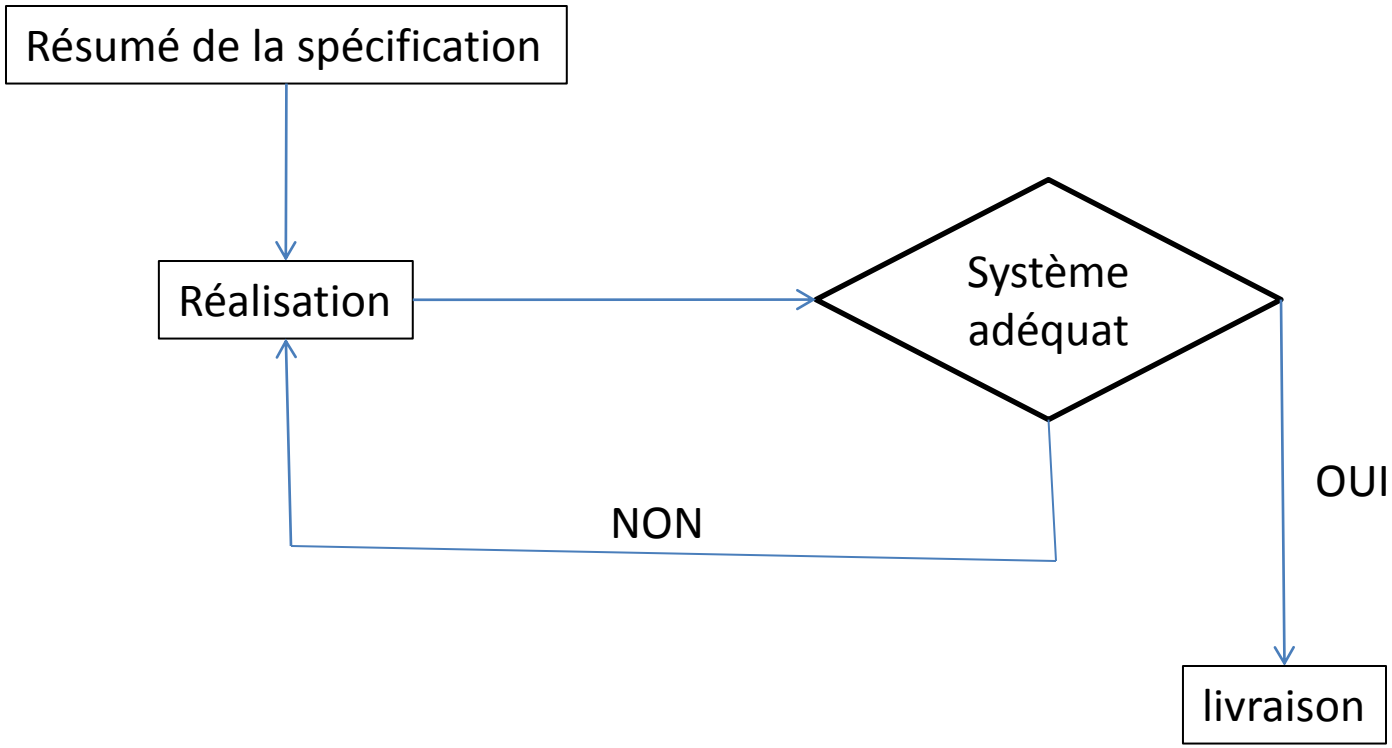
1) Réaliser un résumé de la spécification;

2) Programmer la spécification;

3) **Si** le programme satisfait l'utilisateur **alors**
livraison

Sinon Aller à 2

Approches de développement (2)
la programmation exploratoire(2)



Approches de développement (3)

la programmation exploratoire(3)

- **Limites**:

- Exige des **langages particuliers** (le logiciel change trop);
- Problème de gestion de l'équipe: cette approche nécessite une petite équipe. Donc trop de personnes seront mises en **chômage**.

Approche de **Prototypage**

Approches de développement (4)
Prototypage (1)

- **Objectifs:** **Explorer** les besoins de l'utilisateur à travers un prototype. **Faciliter** la phase d'analyse.

- **Étapes:**

- Découvrir les besoin de l'utilisateur via un prototype.

- Ce prototype sera rejeté (**prototype jetable**), une fois les besoins sont identifiés.

- Le prototype peut évoluer (**prototype évolutif**) pour devenir lui-même le système final

Approches de développement (5)
Prototypage (2)

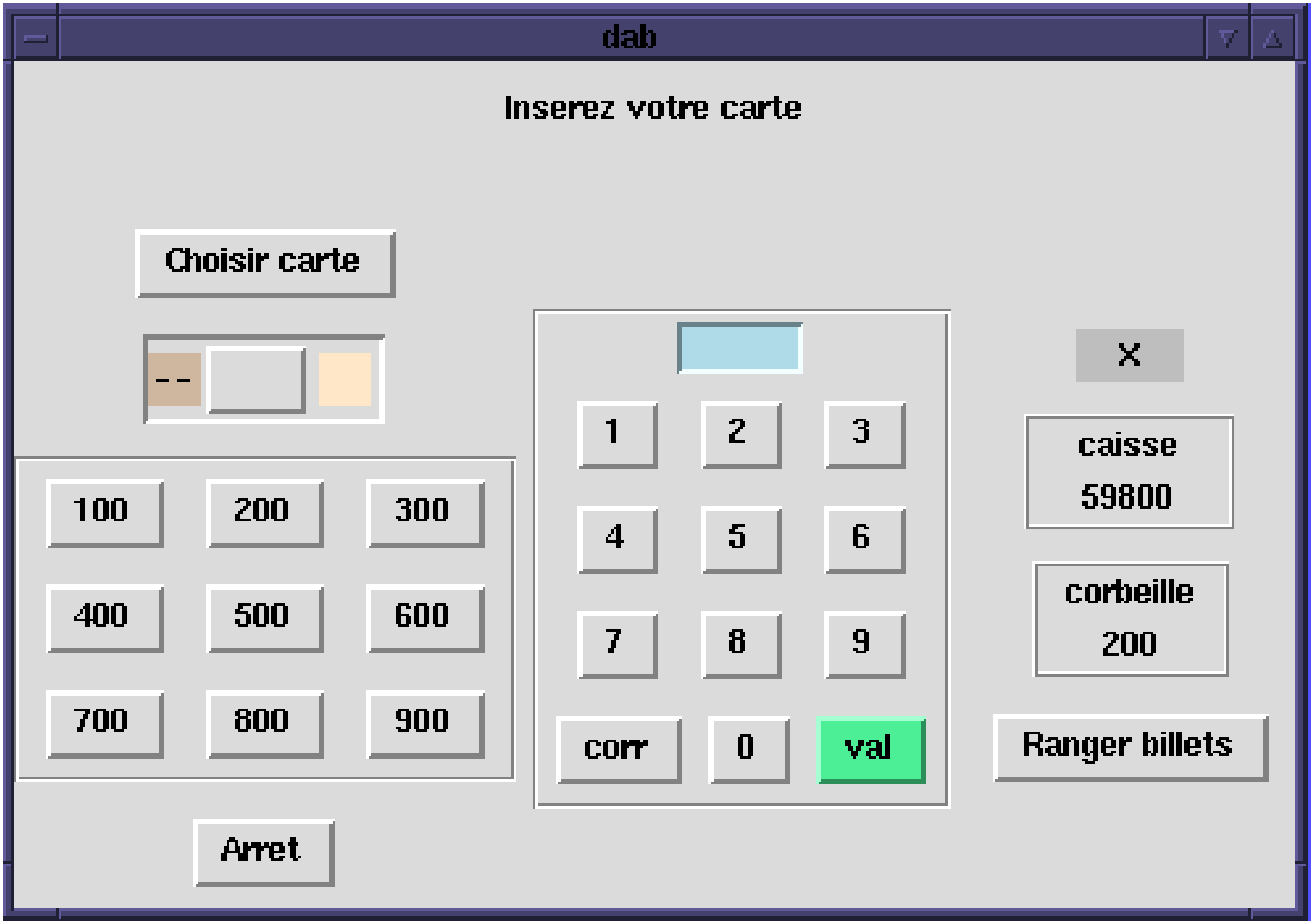
- **Limites**:

- Exige des **outils de prototypage** (dits des langages de 4ème génération)
- Quelques fois **le prix du prototype** devient aussi important que le prix du système lui-même.

- **Exemple et discussion**

Approches de développement (5)
Prototypage (3)

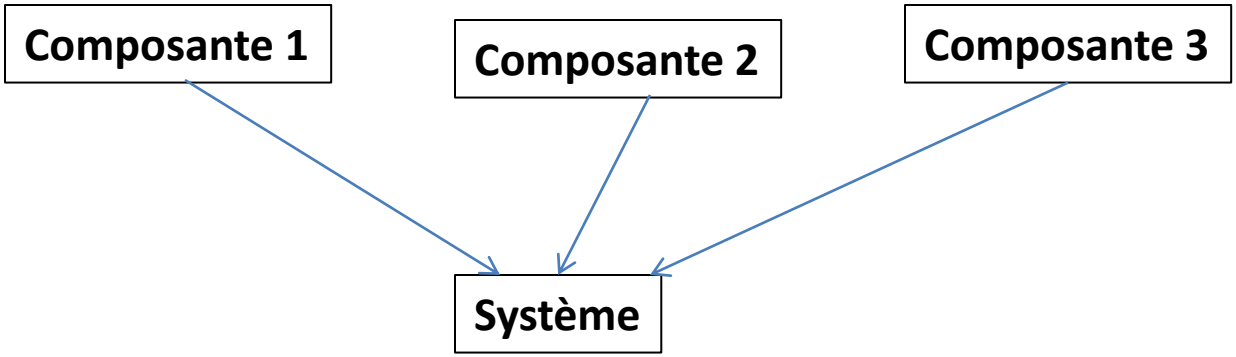
Approches de développement (5)
Prototypage (3)



Approche de la **Réutilisabilité**

Approches de développement (4)
Réutilisabilité (1)

- **Objectif**: minimiser le **coût** et le **temps** de la réalisation.
- **Étapes**: le système n'est pas réalisé à zéro, mais il est **composé de plusieurs** autres composantes déjà existantes.



Approches de développement (5)
Réutilisabilité(2)

- **Limites**:
- Disponibilité et coût des composants?
- Intégration des composants.

- **Exemple et discussion.**

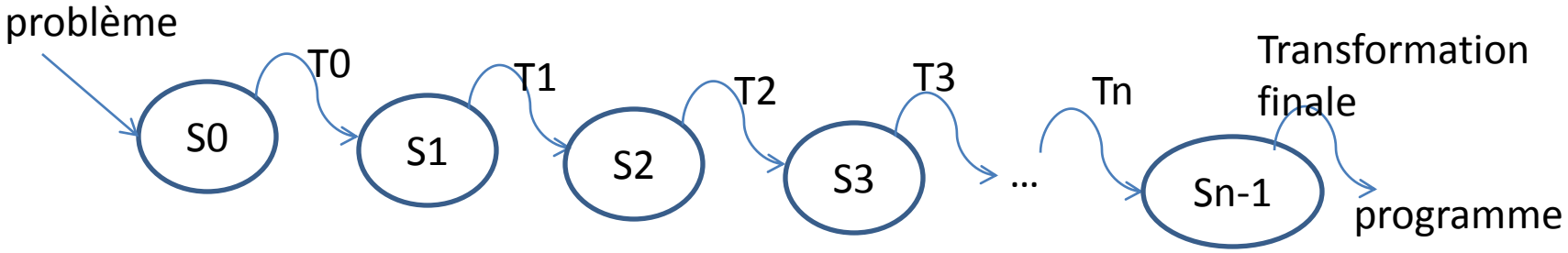
Approche des **Transformations formelles**

Approches de développement (6)
Transformations formelles (1)

- **Objectif**: avoir des systèmes **fiables** et **correctes**. Utiliser la **preuve** de correction. **Minimiser** le coût des tests.

- **Étapes**:
- Réaliser une spécification S0;
- Transformer cette spécification en une autre spécification S1, en appliquant une transformation T0
- Plusieurs transformations seront faites jusqu'à aboutir au programme final.

Approches de développement (7)
Transformations formelles (2)



- **Avantage**:

possibilité d'automatiser tout le processus

Surtout pour les systèmes critiques (la panne est non permise)

- **Limites**:

- Exige des compétences spécifiques: **preuve**, **logique**, **algèbre.**
- Couteux: **rédaction** et **preuves** des spécification.

Pourquoi cette variété d'approche?

- **Pas** d'approches **idéale**;
- À chaque **système** ses **spécificités** et ses **qualités**;
- Les **compétences** de l'équipe;
- Les **outils** et les **moyens** de développement.

Questions

- C’est quoi un prototype? Quels sont les types de prototypes?
- Citer la qualité qui nous pousse souvent à adopter une approche de prototypage?
- Pour un système d’exploitation quelle approches pensez vous plus adéquate?

Modèles des **Incréments**

Autres modèles de gestion (2)
les incréments (1)

- **Objectifs**: minimiser le temps de développement.

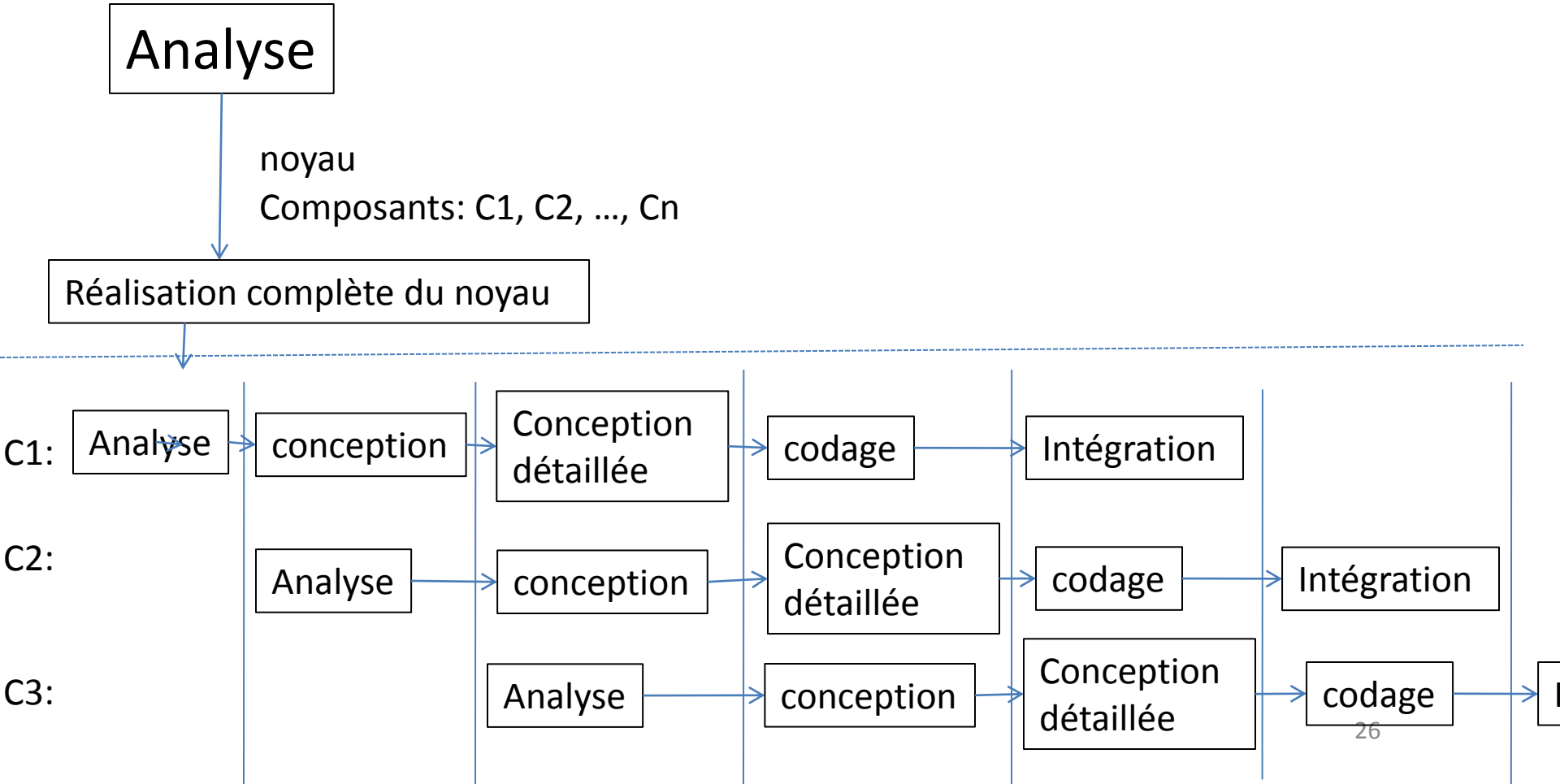
- **principe**: chevaucher (paralléliser) les activités qui peuvent être faites en même temps.

Autres modèles de gestion (2)
les incréments (2)

étapes:

1) Une **première phase d'analyse** et doit identifier le **noyaux du système** et un ensemble de **composants proportionnellement** indépendantes;
2) Réalisation complète du noyaux
3) Le reste des composants se réalisent en parallèle;
4) Chaque composant est intégrée une fois terminé

Autres modèles de gestion (2)
les incréments (3)



Autres modèles de gestion (2)
les incréments (4)

- **Avantage:**

- Possibilité d'avoir des livraisons partielles;
- Bonne gestion de l'équipe

- **Limites**:

- Les systèmes ne sont pas toujours décomposable (facilement);
- L'intégration peut poser souvent des problèmes.

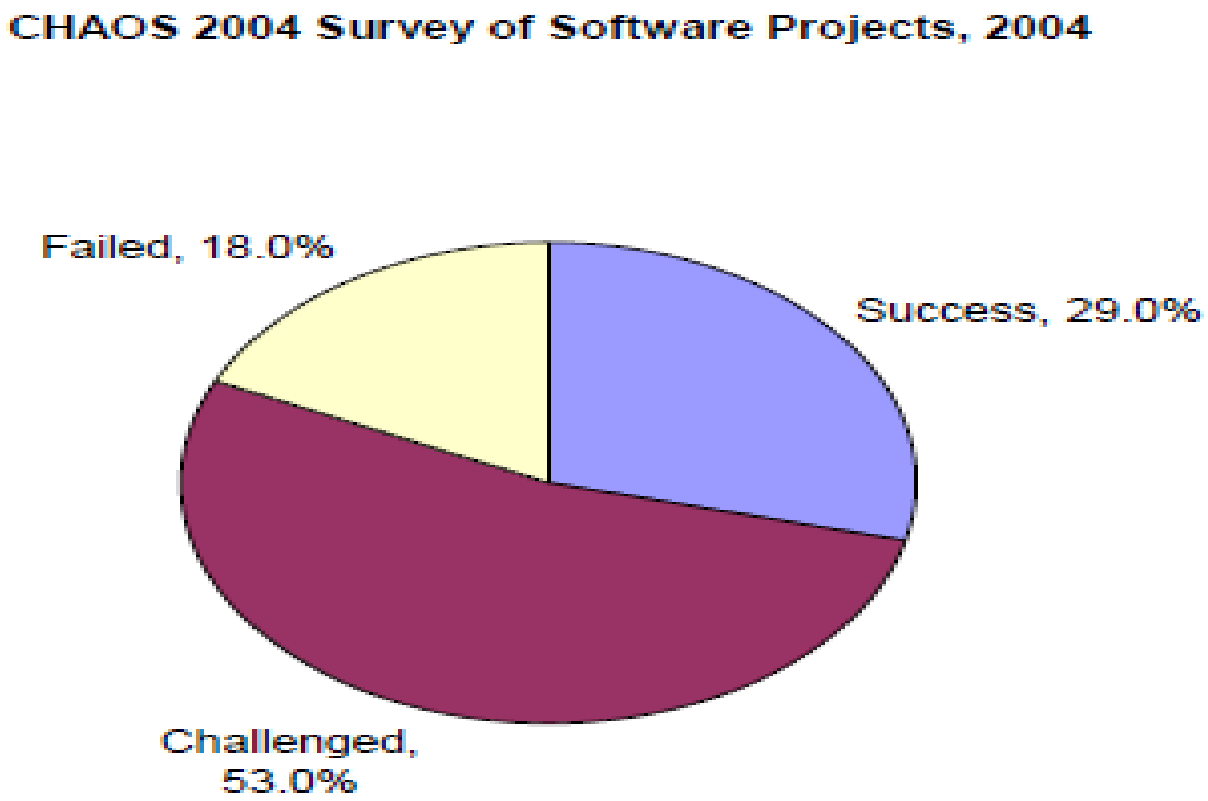
GL dans l'entreprise

- En 1991, au EU ils ont proposé 5 niveaux de maitrise dans le développement de logiciel:

1) **Niveau initial: aléatoire (processus non contrôlé)**
2) **Niveau artisanal: (contrôle de délais)**
3) **Niveau défini: (délais & fiabilité ou délais & coût)**
4) **Niveau géré: (bien contrôlé)**
5) **Niveau optimisé: (recherche dans le processus)**

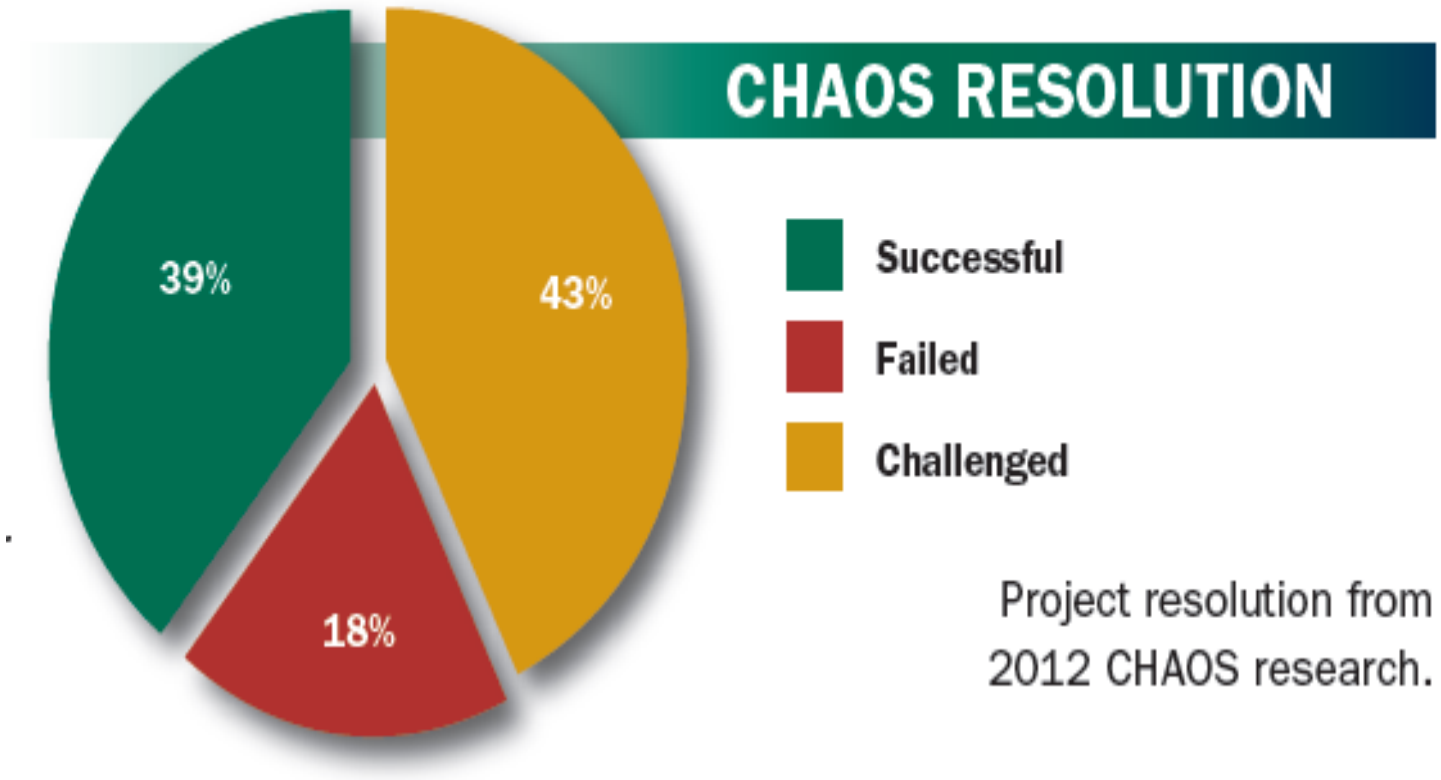
85% niveau 1, 15% niveau, rares (des exceptions) qui sont au niveau 3

GL dans l'entreprise



Challenged: le budget et le calendrier n'est pas respecté

GL dans l'entreprise



GL dans l'entreprise

RESOLUTION

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

GL dans l'entreprise