Chapitre 6

Techniques d'estimation

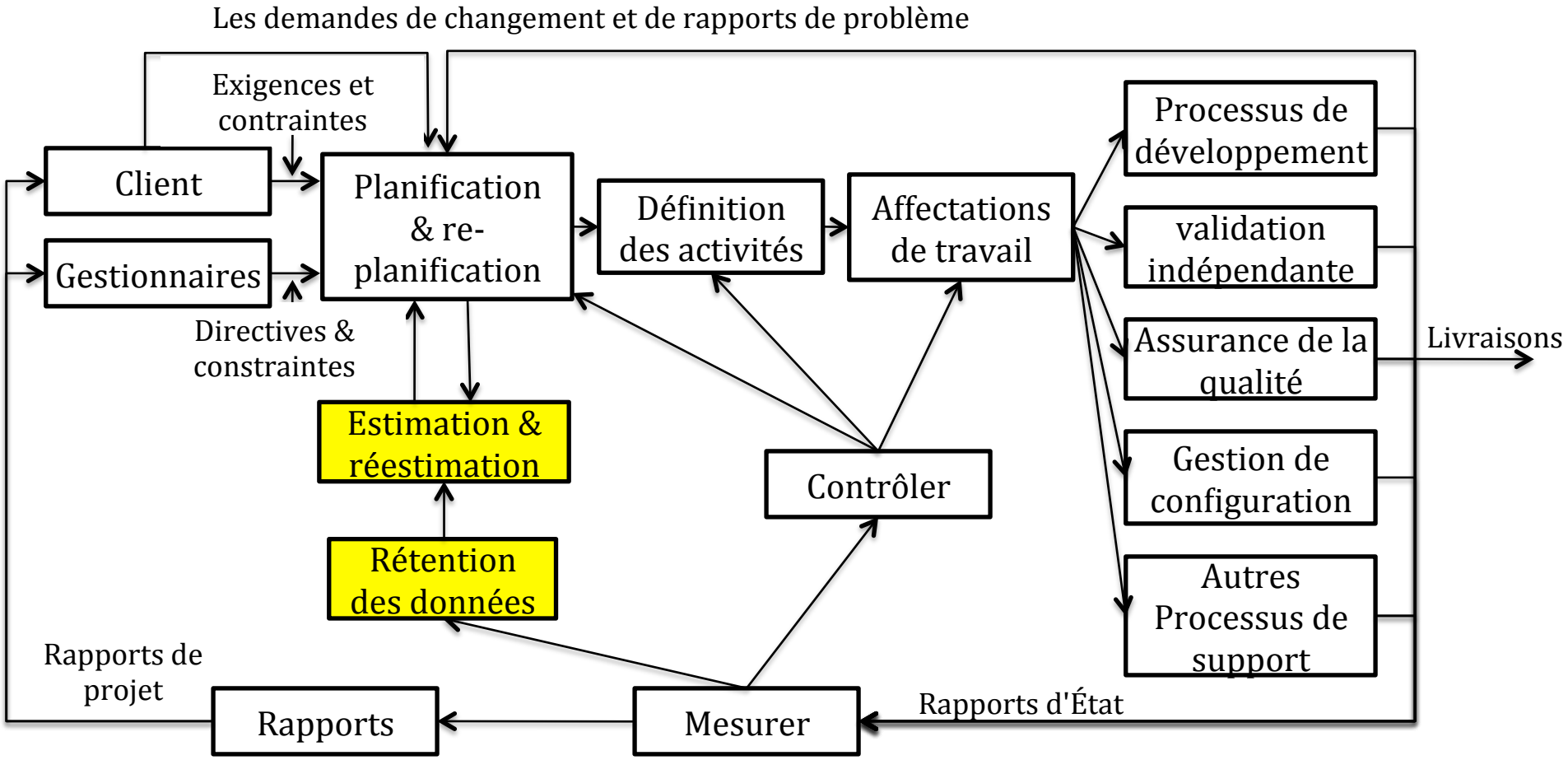
Le contenu est adapté de Richard E. Fairley (2009):
Managing and Leading Software Projects, annotated edition,
Wiley-IEEE Computer Society
Préparé par:
Mazen El-Masri , PMP, M.Sc, ABD
http://ca.linkedin.com/in/mazenmasri
Révisé par:
Jean-Marc Desharnais, ÉTS

Objectifs

- Le rôle de l'estimation
- Trois principes fondamentaux d'estimation
- Le mesure de la taille de projet
- Comment développer une mesure de taille
- Certaines techniques d'estimation non-paramétriques
- *Estimations paramétriques basées sur la théorie et la régression*
- *Comment élaborer, calibrer et d'évaluer l'acceptabilité des modèles d'estimation basée sur la régression*
- *Capacités des outils de l'estimation*
- *Une procédure d'estimation*
- *Un modèle de documentation des estimations*

Le contenu est adapté de Fairley (2009)

Modèle de gestion de projet logiciel



Adapted from Fairley (2009)

Le contenu est adapté de Fairley (2009)

L'objectif de l'estimation

- Déterminer un ensemble de paramètres qui offrent un niveau de confiance élevé qui vous permet de fournir un produit acceptable dans les limites des contraintes du projet.

- Les paramètres et les contraintes à considérer sont:
  - les caractéristiques du produit,
  - attributs de qualité
  - des efforts,
  - les ressources d'autres,
  - calendrier,
  - le budget,
  - la technologie, et
  - une base de l'estimation.

Le contenu est adapté de Fairley (2009)

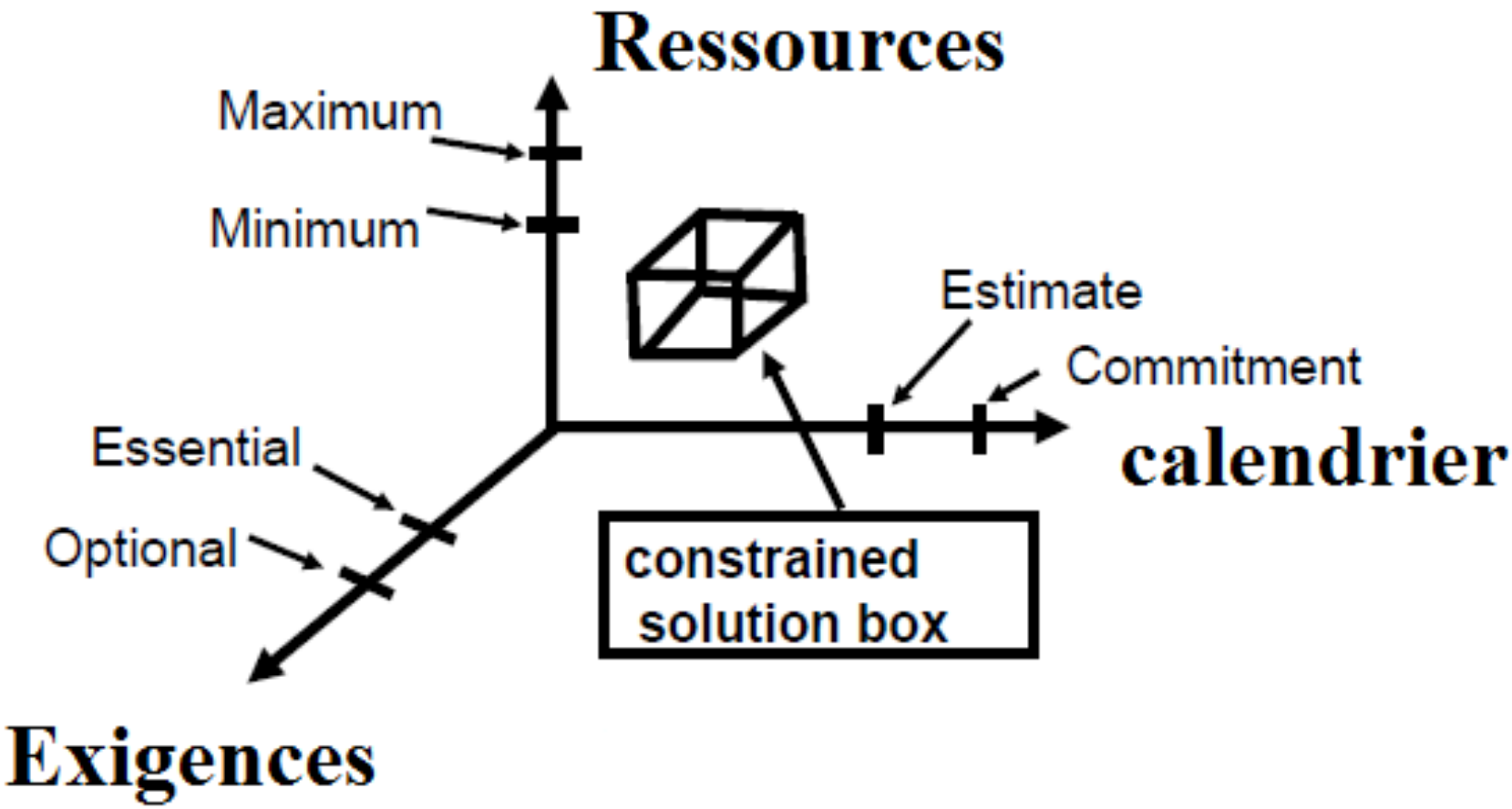
Plan d'estimation (IEEE Standard 1058)

- Un plan pour réaliser des estimations initiales et continues.

  Le plan comprend:
  - les estimations initiales.
  - Les détails du coût du projet, le calendrier, le personnel, exigences, et d'autres ressources
  - Les méthodes, les outils et techniques utilisées
  - Informations historiques utilisées
  - Le niveau de confiance de l'estimateur dans l'estimation
  - Comment ré-estimations périodiques seront faits
  - La fréquence de ré-estimation
  - Le plan de ré-estimation lorsque les exigences ou d'autres conditions du projet changent

Le contenu est adapté de Fairley (2009)

Trois variables fondamentales de l‘estimation



Le contenu est adapté de Fairley (2009)

Agenda

- **Principes fondamentaux de l'estimation**
- Conception de contraintes du projet
- Estimation de la taille du logiciel
- Les techniques d'estimation non-paramétriques
- Des modèles d'estimation paramétrique
  - Modèles d'estimation basée sur la théorie
  - Modèles d'estimation fondées sur la régression estimation Outils
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Principe d'estimation #1: Une définition

- Une estimation est une projection du passé au futur, ajusté pour tenir compte des différences entre passé et futur

- Le passé = des données historiques (base de l'estimation)
  (*vous devez avoir des expériences passées)*
- Le futur = les exigences de logiciel à développer
  = les contraintes du projet

- Les différences = des facteurs d'ajustement (pour tenir compte)

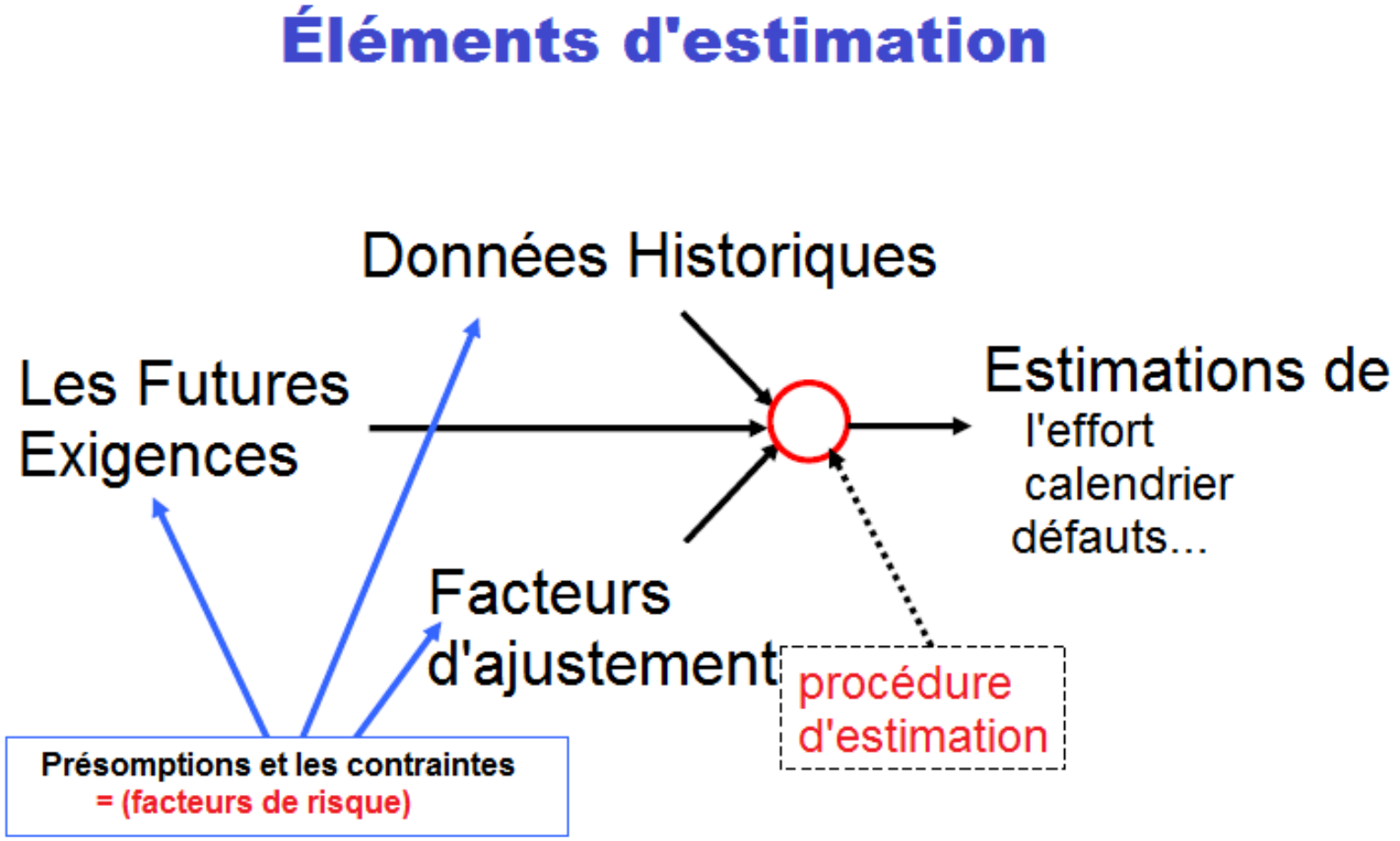
- e.g.: clientèle différente, équipe de développement différents, les différents outils, application différente…

Le contenu est adapté de Fairley (2009)

Exercice: présomption et contrainte

- Indiquez ce qui peut être présomption (assumption) et ce qui peut être une contrainte:
  - futures exigences
  - effort
  - calendrier
  - défauts

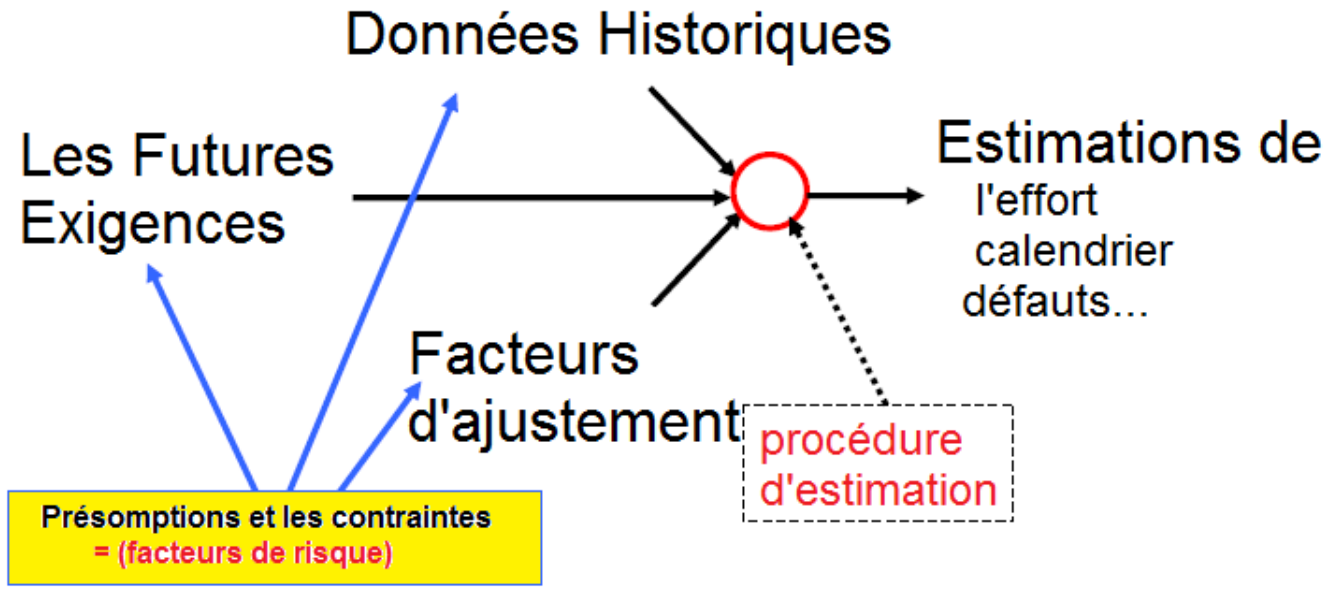
Éléments d'estimation



Le contenu est adapté de Fairley (2009)

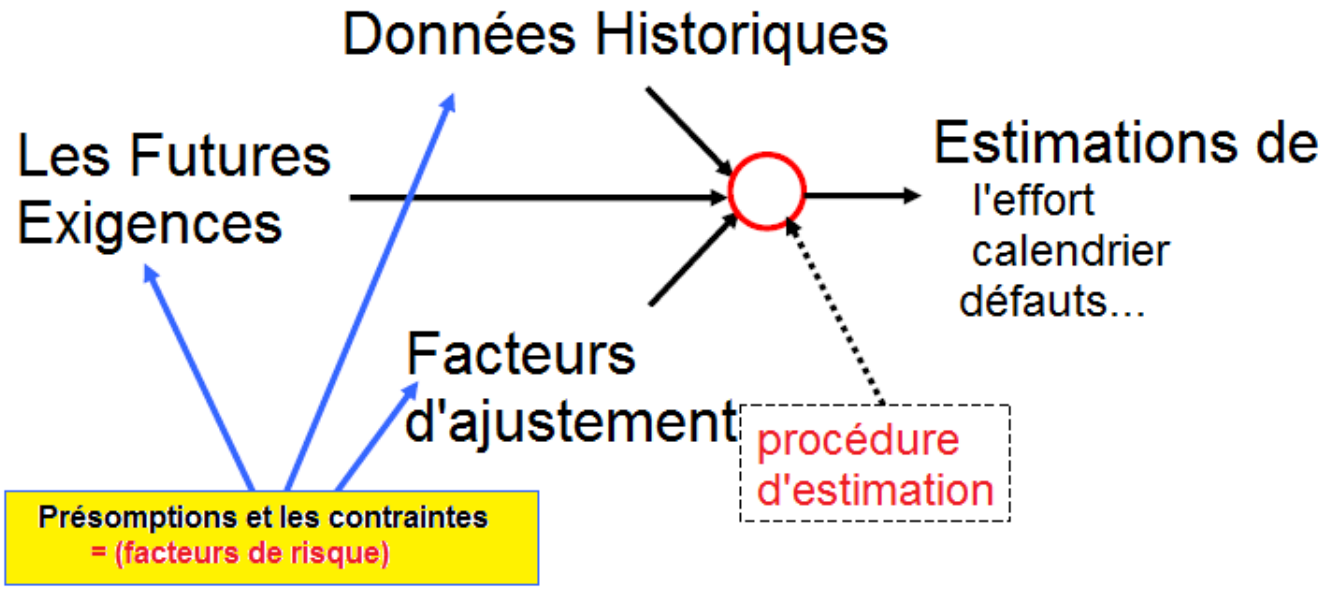
Principe #2: Présomptions et contraintes

- Toutes les estimations sont basées sur un ensemble de présomptions qui doit être réalisé et un ensemble de contraintes qui doivent être satisfaites.



Le contenu est adapté de Fairley (2009)

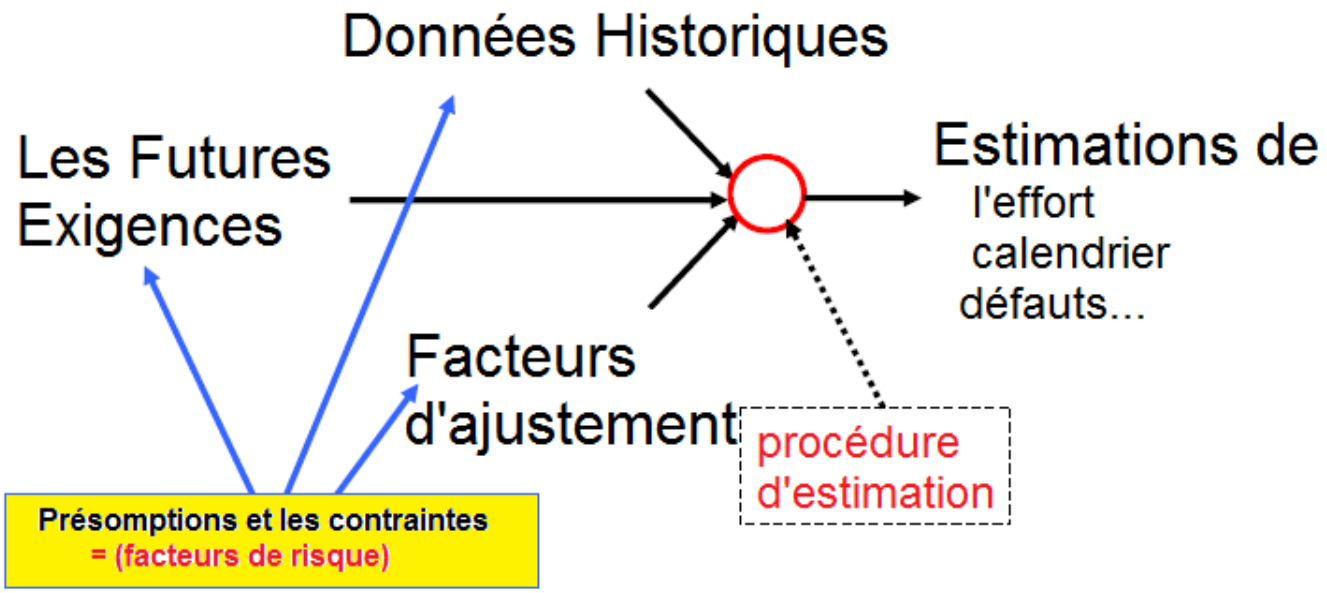
Présomptions et contraintes



- **Une présomption**: une déclaration qui est perçue pour être vrai sans vérifier, ou d'être pouvoir vérifier la véracité de la déclaration. E.g.: productivité sera 500 DSLOC/SM

Le contenu est adapté de Fairley (2009)

Présomptions et contraintes



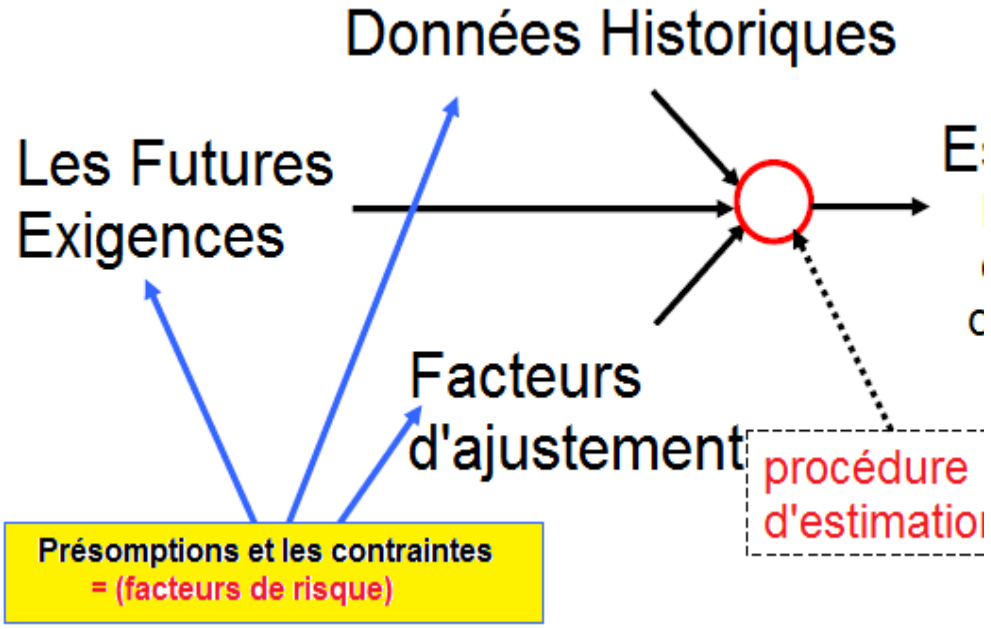
- **Une contrainte:** une condition imposée de l'extérieur qui doivent être respectés. Par exemple, le projet pourrait être contraint à 5 personnes pour 6 mois

Le contenu est adapté de Fairley (2009)

Principe #3: Re-estimation périodique

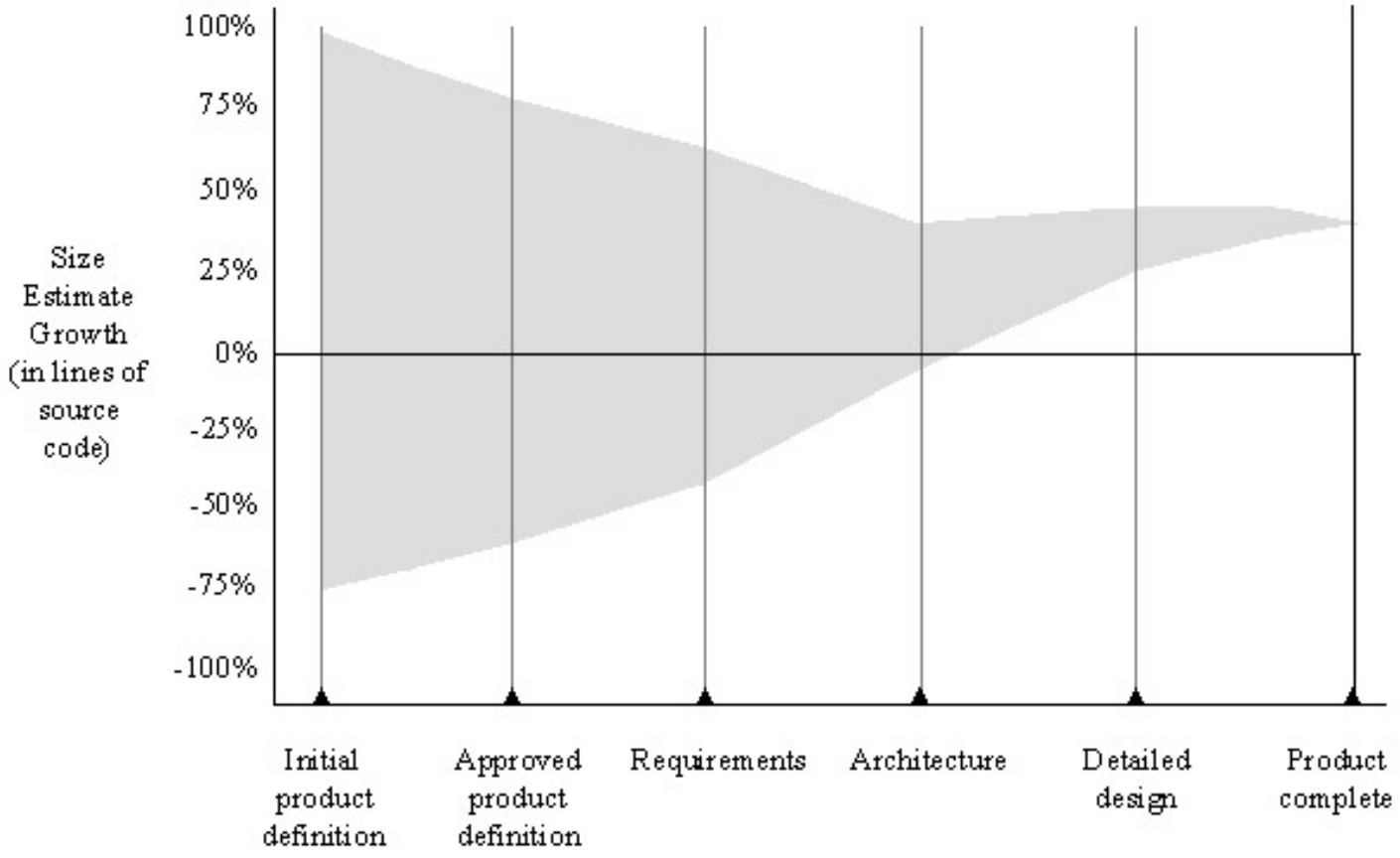
- Les projets doivent être réévaluée périodiquement lorsque la compréhension s'améliore et les paramètres de projet changent

En tant que votre projet évolue, votre compréhensior du produit en cours de développement, les présomptions que vous avez fait, et l'impact des contraintes deviendra plus cl



Le contenu est adapté de Fairley (2009)

Cône de l'incertitude



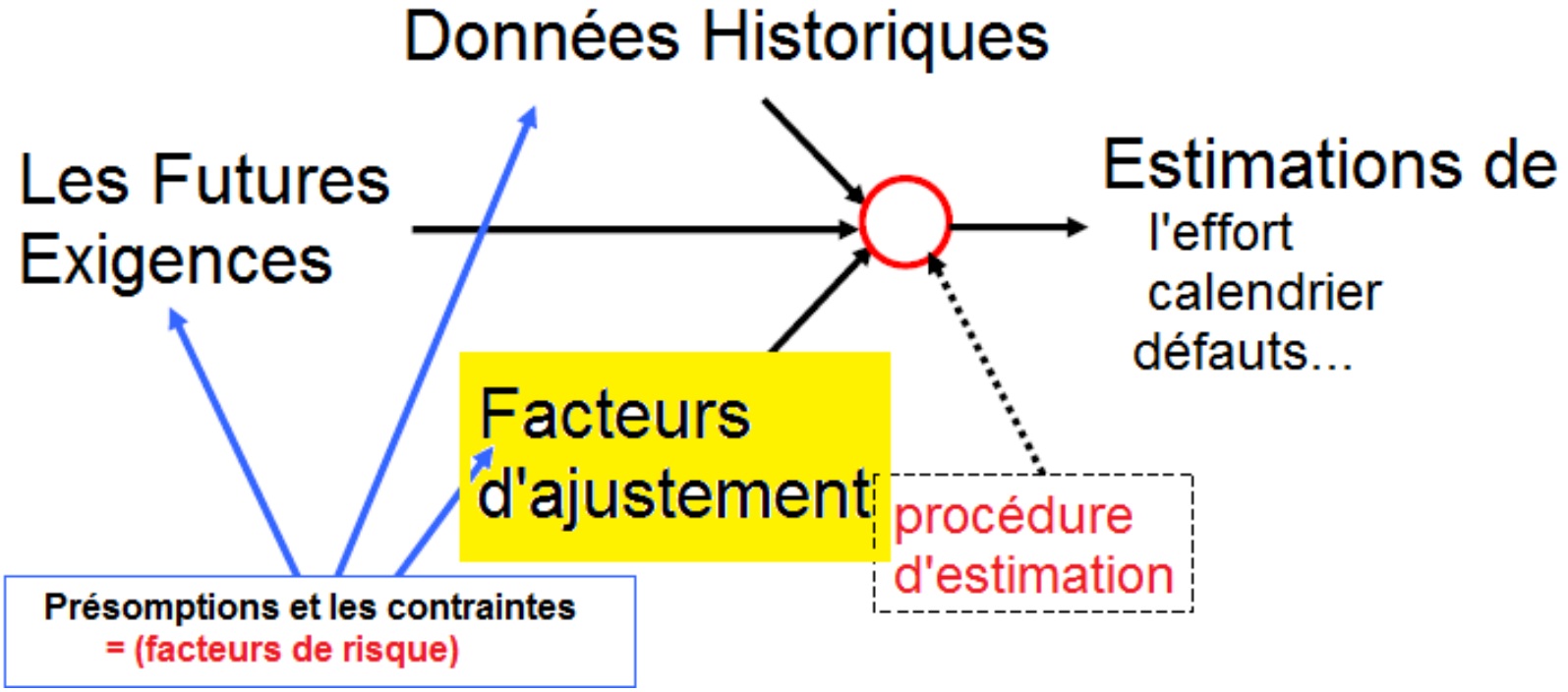
Le contenu est adapté de Fairley (2009)

Agenda

- Principes fondamentaux de l'estimation
- **Conception de contraintes du projet**
- Estimation de la taille du logiciel
- Les techniques d'estimation non-paramétriques
- Des modèles d'estimation paramétrique
  - Modèles d'estimation basée sur la théorie
  - Modèles d'estimation fondées sur la régression
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Facteurs d'ajustement

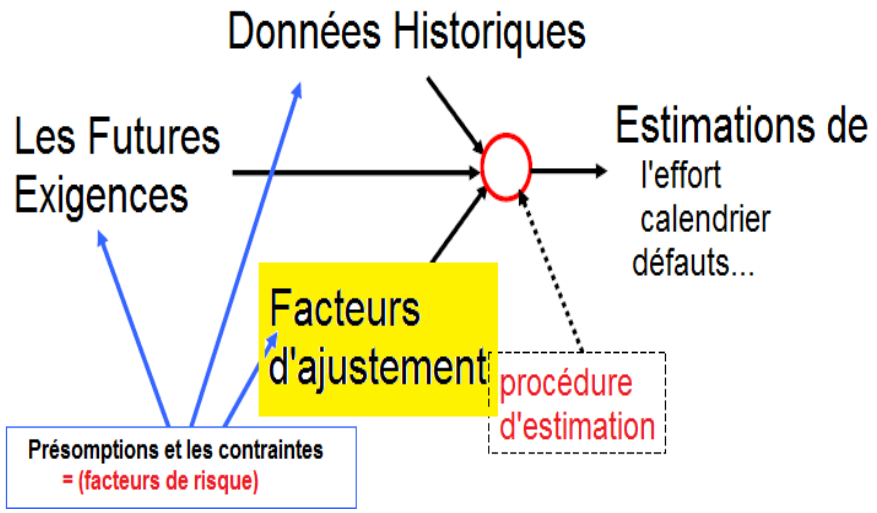


- Le projet futurs pourraient différer des projets précédents d'une manière qui rendra les développeurs plus ou moins productive que la moyenne des projets précédents.

Le contenu est adapté de Fairley (2009)

Facteurs d'ajustement

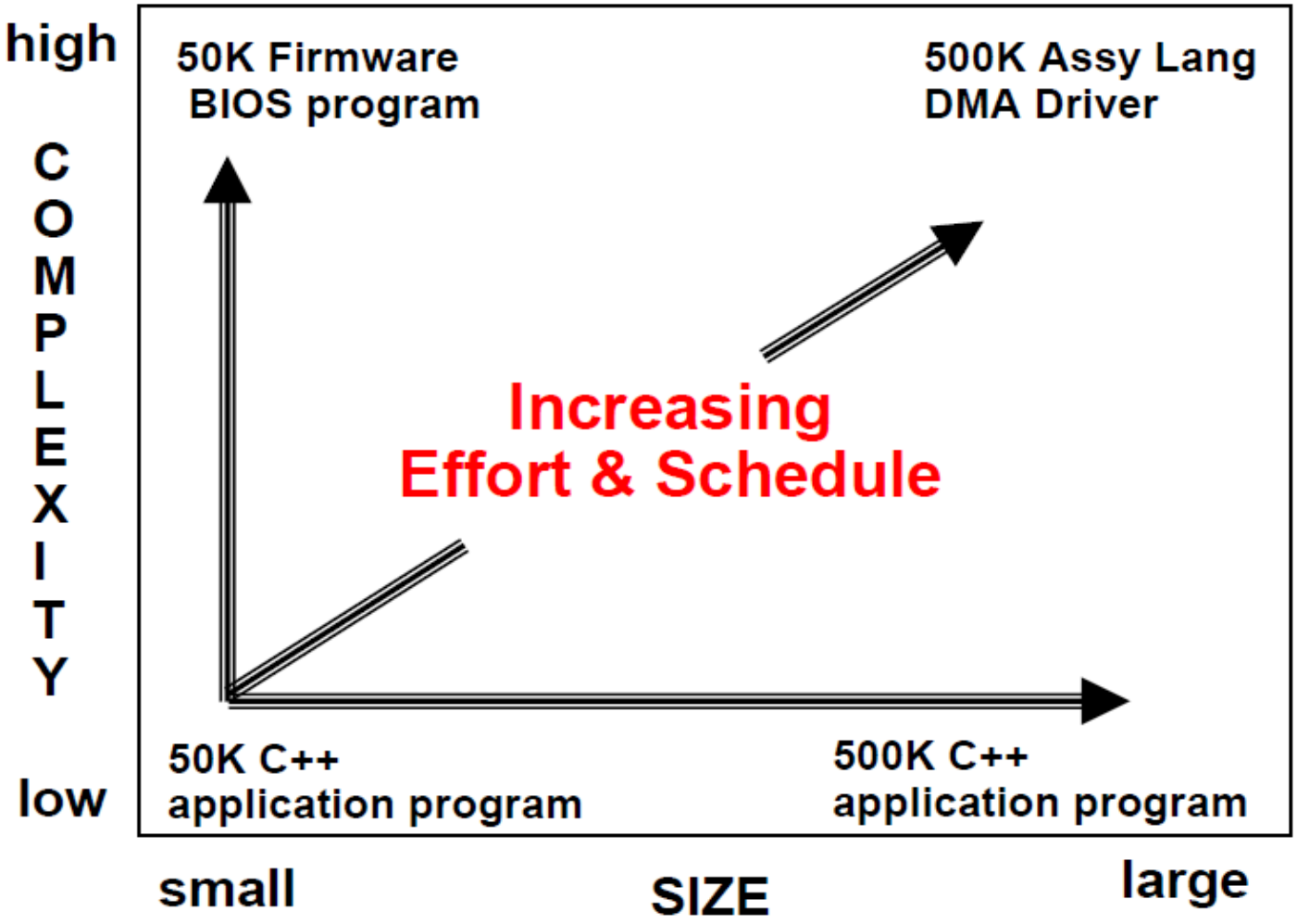
Facteurs d'ajustement typiques incluent:



1. la stabilité des exigences
2. relation avec le client
3. la capacité des développeurs
4. contraintes du calendrier
5. familiarité avec le domaine du problème
6. familiarité avec la plateforme de développement et d'outils logiciels
7. La complexité du problème et la solution

Le contenu est adapté de Fairley (2009)

Facteur d'ajustement: la complexité du produit



Le contenu est adapté de Fairley (2009)

Exercice: facteur qui influence le plus l'effort

- Quel est le facteur qui influence le plus l'effort?
- Pourquoi?

Le contenu est adapté de Fairley (2009)

Agenda

- Principes fondamentaux de l'estimation
- Conception de contraintes du projet
- **Estimation de la taille du logiciel**
- Les techniques d'estimation non-paramétriques
- Des modèles d'estimation paramétrique
  - Modèles d'estimation basée sur la théorie
  - Modèles d'estimation fondées sur la régression
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Estimer la taille du logiciel: certaines façons

- Lignes de code
- Points de fonction
- Nombre de cas d'utilisation
- Classes UML et les relations
- Windows, menus, boutons (Fenêtres, menus, boutons)
- Valves, capteurs, alarmes
- Interruptions, niveaux de priorité, réponses

Estimer la taille de logiciel: Les raisons

- La taille a une forte relation de causalité avec les attributs du projet (l'effort et le calendrier) que les autres attributs
- La taille peut être mesurée de façon plus objective que les autres attributs
- Quelques mesures de la taille peut être estimée avec plus de précision à partir des exigences que les autres attributs de produits peuvent
- Données sur le relation entre taille, l'effort, le calendrier, et d'autres attributs du projets peuvent être collectés à des projets achevés et stockées dans une base de données pour fournir une base historique de l'estimation pour de futurs projets

Le contenu est adapté de Fairley (2009)

Estimer la taille du logiciel: certaines façons

- Lignes de code (ne voit que les aspects techniques, longueur versus taille)
- Points de fonction (basé sur des normes dont ISO/IEC19761 - COSMIC)
- Nombre de cas d'utilisations (trop vague et ne tient pas compte de leur taille, on peut cependant avoir une mesure des cas d'utilisation)
- Classes UML et les relations (idem avec en plus une mesure qui compte des pommes et des oranges)
- Fenêtres, menus, boutons (idem)
- Valeurs, capteurs, alarmes (idem et ne voit qu'une partie du logiciel)
- Interruptions, niveaux de priorité, réponses (idem au précédent)

Le contenu est adapté de Fairley (2009)

Points de fonction FPA versus COSMIC

- FPA
- COSMIC

- Avantages et inconvénients

Le contenu est adapté de Fairley (2009)

Estimer la taille de logiciel: Lignes de code

Problèmes avec utilisant des lignes de code:

- Il est difficile d'estimer de lignes de code au début d'un projet,
- il est difficile de relier les modifications aux exigences avec des changements dans les lignes de code estimé
- Calcul de la productivité des lignes de code générées par le programmeur/ mois peut encourager les programmeurs d'écrire beaucoup de lignes de code de mauvaise qualité
- Les méthodes de développement modernes rendre la relation entre les lignes de code et le projet attributs moins pertinent et moins précis que dans le passé
- La mesure tient compte que de l'aspect technique du projet

Le contenu est adapté de Fairley (2009)

Explication de COSMIC

- Processus fonctionnel
- Entrée, lecture, écriture, sortie
- Calcul des points
- Exemples
- Exercice

ISBSG

- Ce qu'est ISBSG
- Utilisation d'ISBSG
- Limites

Le contenu est adapté de Fairley (2009)

Agenda

- Principes fondamentaux de l'estimation
- Conception de contraintes du projet
- Estimation de la taille du logiciel
- **Les techniques d'estimation non-paramétriques**
- Des modèles d'estimation paramétrique
  - Modèles d'estimation basée sur la théorie
  - Modèles d'estimation fondées sur la régression
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Techniques d'estimation non paramétriques

- WBS - CPM - PERT
- Analogie
- Règle du pouce (rule of thumb)
- Jugement d'experts
- Estimation Delphi

Le contenu est adapté de Fairley (2009)

Techniques d'estimation non paramétriques: WBS-CPM-PERT

1. Estimer l'effort pour chaque tâche dans la WBS
2. Calculer l'effort global pour les tâches
3. Spécifiez le tâche prédécesseur et les successeur pour chaque tâche.
4. Déterminer le chemin critique (ou chemins) pour trouver le calendrier optimal
5. Utilisez l'approche PERT pour estimer la probabilité de réaliser différentes dates de calendrier
6. ajuster calendrier et/ou de ressources pour atteindre un profil de dotation acceptable

Le contenu est adapté de Fairley (2009)

Techniques non-paramétriques: Estimation par analogie

- Analogie est l'une des techniques plus utilisé
- Analogie simple:
  - basé sur le exigences, il semble qu'un travail similaire a eu 5 personnes 6 mois pour terminer
  - quelques exigences sont différentes, donc nous allons planifier le projet pour 5 personnes, 8 mois
- Analogie sophistiqué :
  - décrire les attributs de votre projet et produit
  - rechercher des projets similaires dans votre base de données historique des projets passés
  - Utilisez les projets similaires comme base de l'estimation
  - faire des ajustements nécessaires

Le contenu est adapté de Fairley (2009)

Techniques non-paramétriques: Estimation par analogie

- Analogie est l'une des techniques plus utilisé
- Analogie simple:
  - basé sur le exigences, il semble qu'un travail similaire a eu 5 personnes 6 mois pour terminer
  - quelques exigences sont différentes, donc nous allons planifier le projet pour 5 personnes, 8 mois
- Analogie sophistiqué :
  - décrire les attributs de votre projet et produit
  - rechercher des projets similaires dans votre base de données historique des projets passés
  - Utilisez les projets similaires comme base de l'estimation
  - faire des ajustements nécessaires

Le contenu est adapté de Fairley (2009)

Techniques non-paramétriques: règle du pouce générale

- Règle du pouce ou « Rule of Thumb » peuvent être basées sur des moyennes de l'industrie ou des données locales.

- Exemple:
  - Règle général de productivité: Notre productivité est généralement de 500 LOC / SM
  - Règles général de qualité:
    - 20 défauts par KLOC
    - Taux de capture de défaut est 90%
  - Règle général d'horaire: Il prend généralement environ un mois pour faire des tests du système final

Le contenu est adapté de Fairley (2009)

Techniques non-paramétriques: Estimation Delphi

1. Les experts reçoivent l'information sur laquelle ils vont fonder l'estimation
2. travaillent seuls et soumettre leurs estimations et leurs justifications à la coordonnatrice
3. Le coordinateur prépare un rapport anonyme qui contient les estimations et les justifications de chaque estimateur.
   - Il donne le rapport de chaque estimateur
   - demande à chacun de soumettre une deuxième estimation
4. La procédure continue jusqu'à ce que les estimations stabilisent
   - si il y a des petite disparité dans les estimations stabilisées, ils peuvent servir comme la gamme des estimations
   - si il y a des grande disparité, les estimateurs se rencontrer pour discuter et résoudre leurs désaccords

Le contenu est adapté de Fairley (2009)

Agenda

- Principes fondamentaux de l'estimation
- Conception de contraintes du projet
- Estimation de la taille du logiciel
- Les techniques d'estimation non-paramétriques
- **Des modèles d'estimation paramétrique**
  - Modèles d'estimation basée sur la théorie
  - Modèles d'estimation fondées sur la régression
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Modèles d'estimation paramétriques

- Les estimations sont calculées en utilisant les équations qui ont certains paramètres d'entrée, tels que :

$$\text{Effort} = f(P1, P2, \ldots Pn)$$

- Par exemple:

$$\text{Effort} = a \times \text{taille}^{b} \times \text{temps}^{c}$$

Lorsque:

- La taille est le taille du produit estimée
- L'effort est en mois-personne
- Le temps est en mois
- a, b et c sont des constantes déterminé par des circonstances locales

Agenda

- Principes fondamentaux de l'estimation
- Conception de contraintes du projet
- Estimation de la taille du logiciel
- Les techniques d'estimation non-paramétrique
- Des modèles d'estimation paramétriques
  - **Modèles d'estimation basée sur la théorie**
  - Modèles d'estimation fondées sur la régression
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

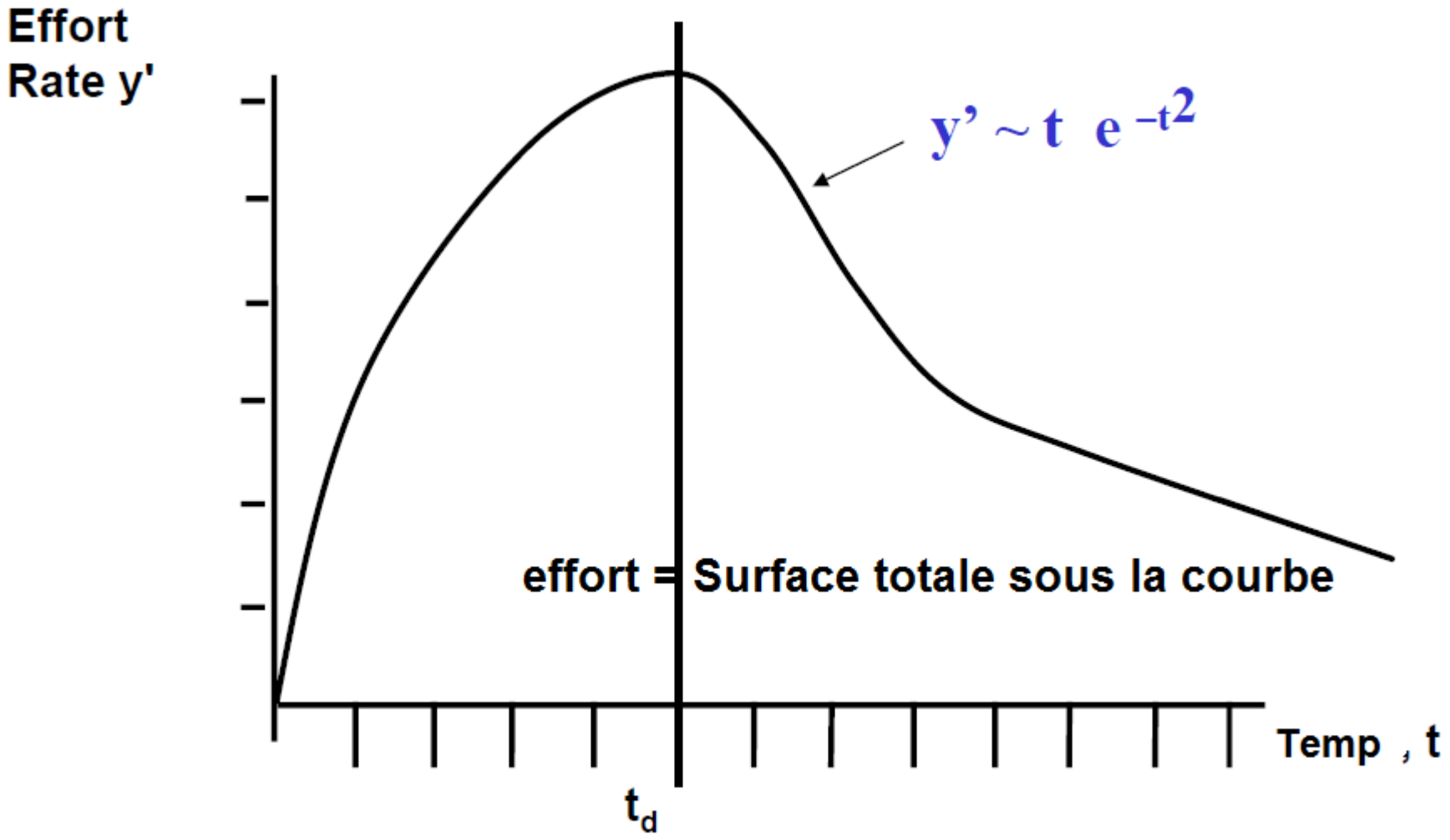
Le contenu est adapté de Fairley (2009)

Modèles d'estimation basée sur la théorie: SLIM

- SLIM: Lifecycle Management Software
- SLIM est un outil d'estimation commerciales agréées par la Compagnie QSM
- Le modèle d'estimation est basée sur deux équations à deux inconnues (effort et durée)
- Solutions simultanées des équations pour produire des estimation comme des combinaisons d'efforts et de temps

Le contenu est adapté de Fairley (2009)

SLIM: L'équation de l'effort de Norden-Rayleigh



Le contenu est adapté de Fairley (2009)

Les équations originales de SLIM

- Deux équations à deux inconnues (E et T)
  - E: effort en mois-personne
  - T: calendrier en mois

Èquations #1: $E = MBI \times T^3$

(basé sur l'équation de Rayleigh-Norden)

Èquations #2: $E = (Size/PI)^3 \times T^{(-4)}$

(basée sur l'équation du logiciel de Putnam)

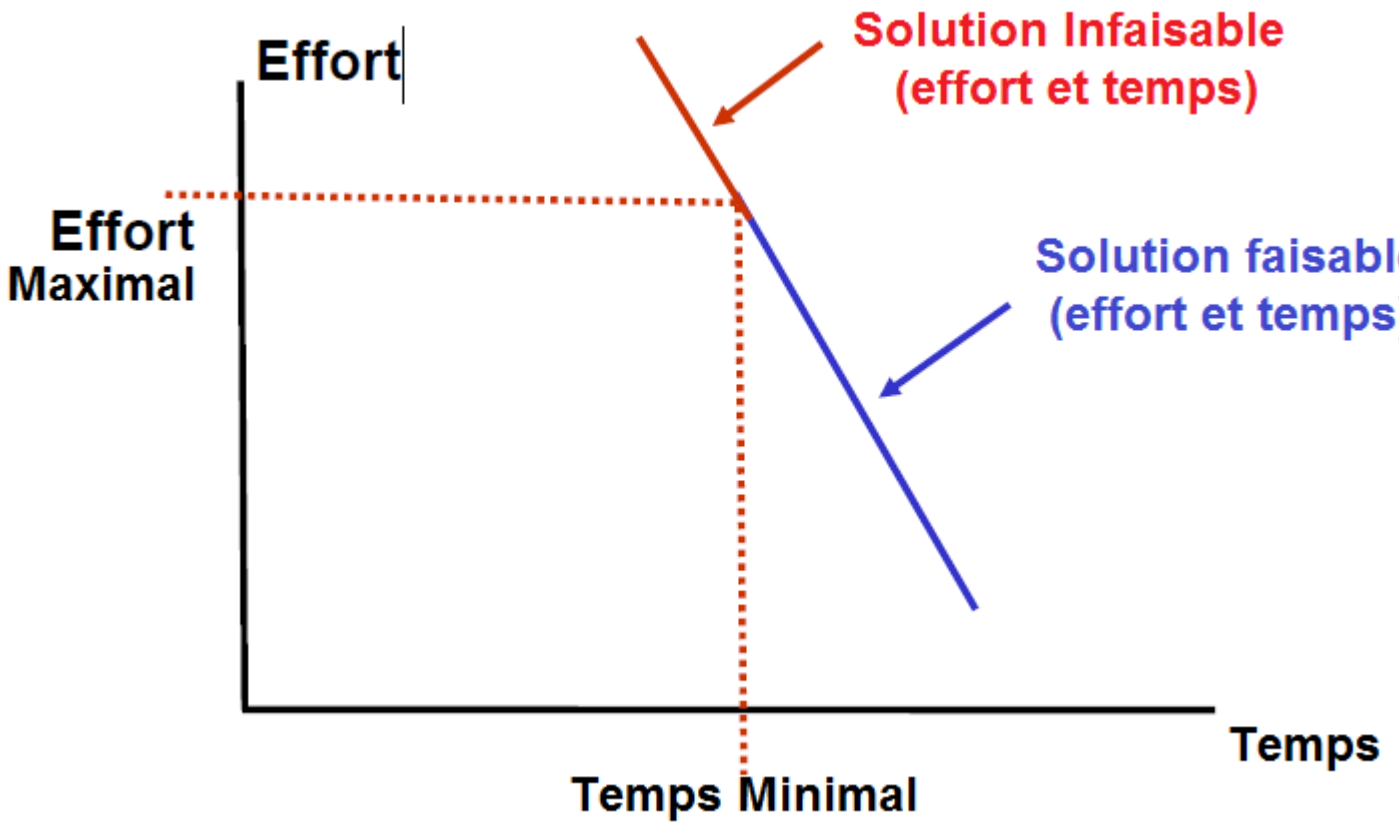
Les équations originales de SLIM

Èquations #1: $E = MBI \times T^3$

Èquations #2: $E = (Size/PI)^3 \times T^{(-4)}$

- Les paramètres d'entrée des équations sont les suivantes:
  - **Taille**: taille du produit estimée – exprimée en DSLOC, FPs…
  - **MBI** (Manpower Build-up Index): reflète le taux estimé de l'accumulation de personnel de projet;
  - **PI**: l'indice de productivité. Les données locales peuvent être utilisés pour calibrer le paramètre PI. Les valeurs moyennes de l'industrie pour des types de produits différents peuvent être utilisés
- La sortie: les combinaisons d'efforts et temps

Solution simultanée des équations de SLIM



Le contenu est adapté de Fairley (2009)

Agenda

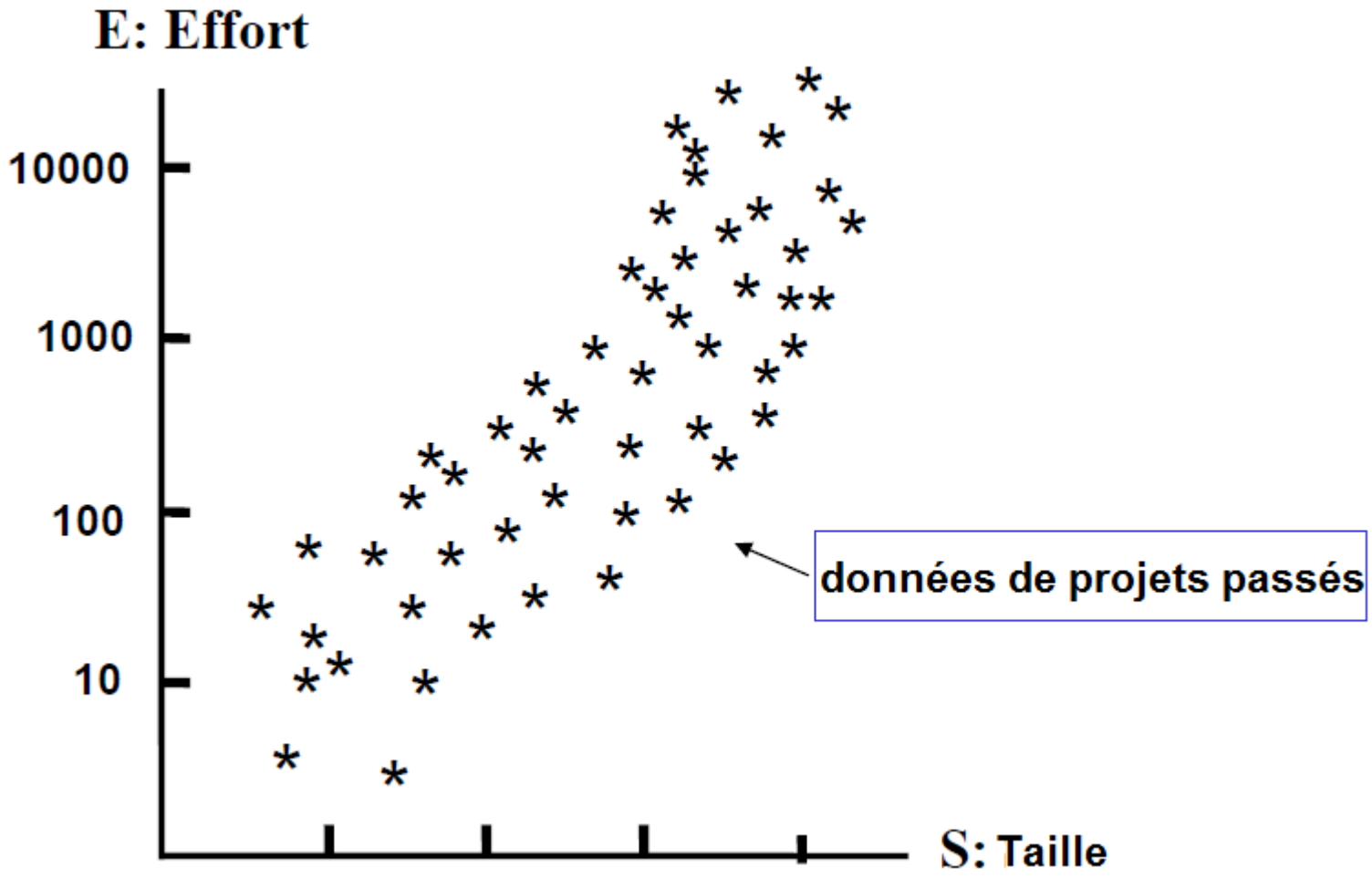
- Principes fondamentaux de l'estimation
- Conception de contraintes du projet
- Estimation de la taille du logiciel
- Les techniques d'estimation non-paramétriques
- Des modèles d'estimation paramétrique
  - Modèles d'estimation basée sur la théorie
  - **Modèles d'estimation fondées sur la régression**
- Outils d’estimation
- Estimation des ressources du cycle de vie, effort et le coût
- Une procédure d'estimation
- Un modèle de documentation des estimations

Le contenu est adapté de Fairley (2009)

Équation de régression typiques

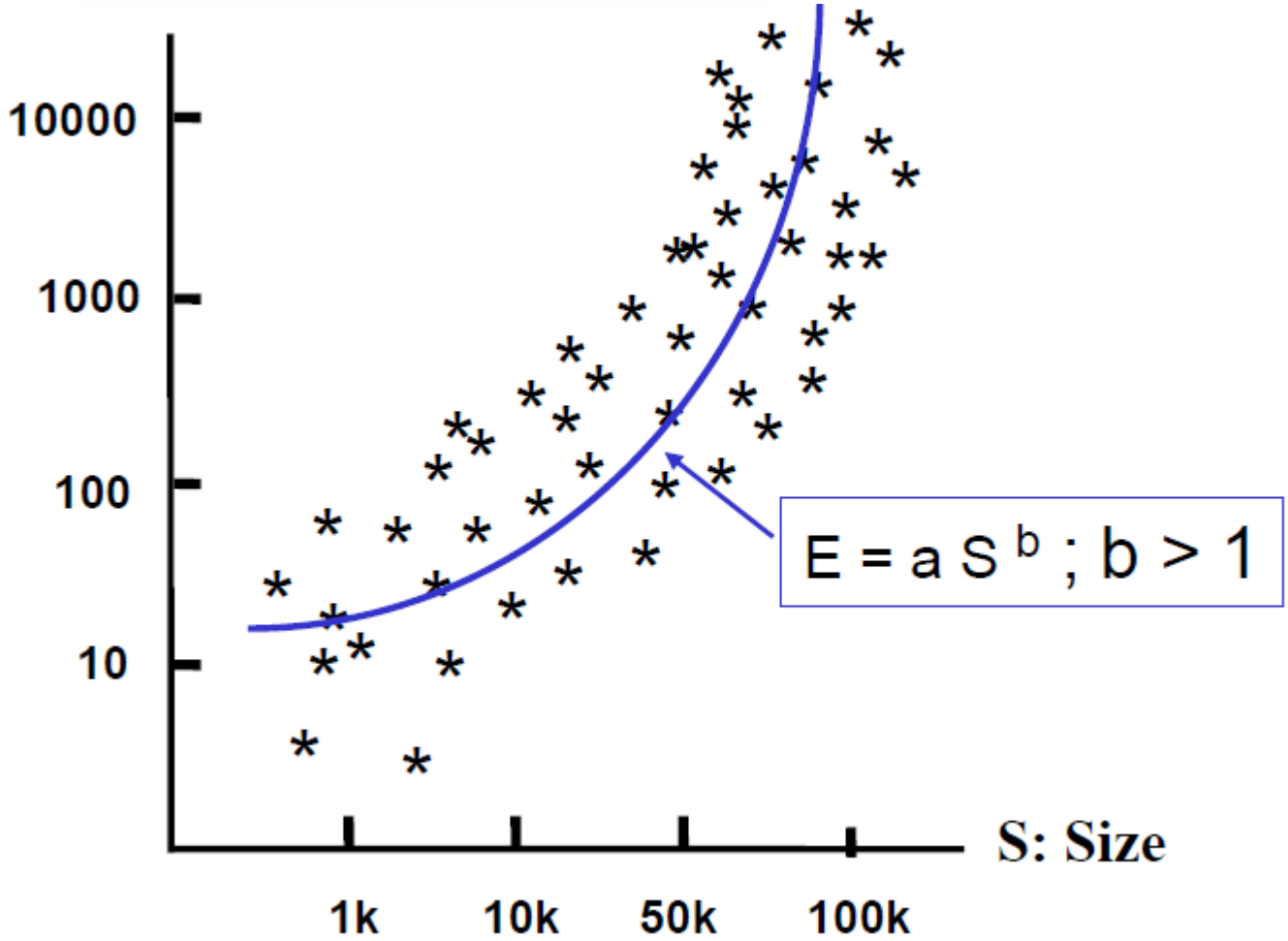
- **Effort = a*(Taille)^{b} * CPLX**
- **CPLX** (Problem and solution complexity) = La complexité du problème et la solution
- La complexité du problème et la solution **(CPLX)** est un **facteur d'ajustement**
- Les données historiques sont résumées dans l'équation exponentielle: **a*(Taille)^{b}**
- e.g.: Effort = 3,2 (50) 1,25 * 1,2 = 510 mois-personne

Un diagramme de dispersion de l'effort par rapport le taille des projets passés



Le contenu est adapté de Fairley (2009)

Une équation de régression basée sur un diagramme de dispersion



Le contenu est adapté de Fairley (2009)

COCOMO : le modèle de régression le plus connu

- COCOMO : COnstructive COst Model
  : une famille de modèles d'estimation

- COCOMO81 : développé par Dr Barry Boehm tout à TRW vers 1980.

  Documenté en « Software Engineering Economics » par Barry Boehm, Prentice Hall, 1981

  Mise à jour en 1987 pour le développement progressif

- COCOMO II : a été publié en 1998 et mis à jour en 2000
  Documenté dans « Software Cost Estimation with COCOMO II » par Boehm, et al., Prentice Hall, 2000

Une équation de l'Effort avec EMs

AFs = facteurs d'ajustement = facteurs de coûts (Cost Drivers)

EMs = Les valeurs attribuées aux facteurs de coûts sont appelés multiplicateurs d'effort (Effort Multipliers)

Une équation de l'Effort avec EMs

- EMs = Effort multiplier (multiplicateurs d'effort)
- Effort = 3.3 * (Size)$^{1.25}$ * EM_1 * EM_2 . . . * Em_n

  ou
- Effort = 3.3 * (Size)$^{1.25}$ * EAF
- EAF (facteur d'ajustement des efforts) = π (EMs)
- π (EMS) = le produit de multiplicateurs d'effort
- E.g:
  - Em_1 = CPLX (la complexité du produit) = 0.9
  - Em_2 = FIER (la fiabilité requises) = 1.4
  - Em_3 = SCED (contrainte horaire) = 1.2
  - ➔ EAF = CPLX x FIER x SCED = **1.3**
  - ➔ Effort = 3.3 * (Size)$^{1.25}$ * 1.3 = ?

COCOMO81: Certains équations d'effort et Calendrier

- $SM = 3,6 * (KDSI)^{1,20} * EAF$
- $TDEV = 2,5 * (SM)^{0,32}$

lorsque

- SM = effort personnel en mois (Staff Month)
- TDEV = 1000 d'instructions source livré (DSLOC)
- AEF = le facteur d'ajustement de l'effort
- TDEV = le calendrier de développement

COCOMO81: Quinze COCOMO81 inducteurs de coût

Cost Drivers		Ratings					
		Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes							
RELY:	Required Software Reliability	0.75	0.88	1.00	1.15	1.40	
DATA:	Data Base Size		0.94	1.00	1.08	1.16	
CPLX:	Product Complexity	0.70	0.85	1.00	1.15	1.30	1.65
Computer Attributes							
TIME:	Execution Time Constraint			1.00	1.11	1.30	1.65
STOR:	Main Storage Constraint			1.00	1.06	1.21	1.56
VIRT:	Virtual Machine Volatility*		0.87	1.00	1.15	1.30	
TURN:	Computer Turnaround Time		0.87	1.00	1.07	1.15	
Personnel Attributes							
ACAP:	Analyst Capability	1.46	1.19	1.00	0.86	0.71	
AEXP:	Applications Experience	1.29	1.13	1.00	0.91	0.82	
PCAP:	Programmer Capability	1.42	1.17	1.00	0.86	0.70	
VEXP:	Virtual Machine Experience*	1.21	1.10	1.00	0.90		
LEXP:	Programming Language Experience	1.14	1.07	1.00	0.95		
Project Attributes							
MODP:	Use of Modern Programming Practices	1.24	1.10	1.00	0.91	0.82	
TOOL:	Use of Software Tools	1.24	1.10	1.00	0.91	0.83	
SCED:	Required Development Schedule	1.23	1.08	1.00	1.04	1.10	

Le contenu est adapté de Fairley (2009)

COCOMO81: Un exemple

Estimated size = 50,000 KDSI

Cost Driver	Situation	Rating	Effort Multiplier
RELY	Local Use of System, No Serious Recovery Problems	Nominal	1.00
DATA	20,000 Bytes	Low	0.94
CPLX	Communications Processing	Very High	1.30
TIME	Will Use 70% of Available Time	High	1.11
STOR	45K of 64K Store (70%)	High	1.06
VIRT	Based on Commercial Microprocessor Hardware	Nominal	1.00
TURN	Two-Hour Average Turnaround Time	Nominal	1.00
ACAP	Good Senior Analysts	High	0.88
AEXP	Three Years	Nominal	1.00
PCAP	Good Senior Programmers	High	0.86
VEXP	Six Months	Low	1.10
LEXP	Twelve Months	Nominal	1.00
MODP	Most Techniques in Use Over One Year	High	0.91
TOOL	At Basic Minicomputer Tool Level	Low	1.10
SCED	Nine Months	Nominal	1.00
	Effort Adjustment Factor (Product of Effort Multipliers)		**1.17**

Le contenu est adapté de Fairley (2009)

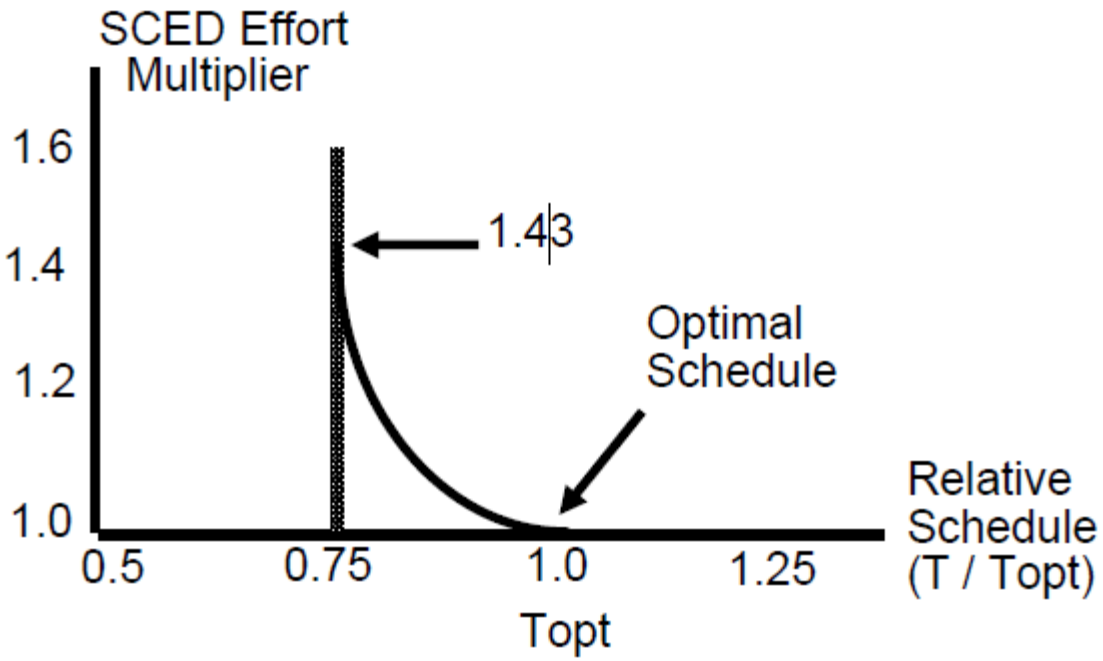
COCOMO81: Un exemple

- Effort = $2.6 * (KDSI)^{1,20}$
- ➔ Effort = $2.6 * (50)^{1,20}$ = 306.1 SM
- EAF = 1.17
- ➔ Effort ajusté = 306.1 SM x 1.17 = 358.2 SM

- TDEV = $2,5 * (SM)^{0,32}$
- ➔ Schedule = $2.5 * (358.2)^{0,32}$ = 16.4 mois

- Niveau Personnel Moyenne = 358.2 SM /16.4 mois ~ 22 pers
- Productivité = 50,000 / 358.2 ~ 140 LOC/SM
- Coûts = $10,000 per SM * 358.2 SM = $ 3.58 M

COCOMO81: Le EM pour la compression Horaire



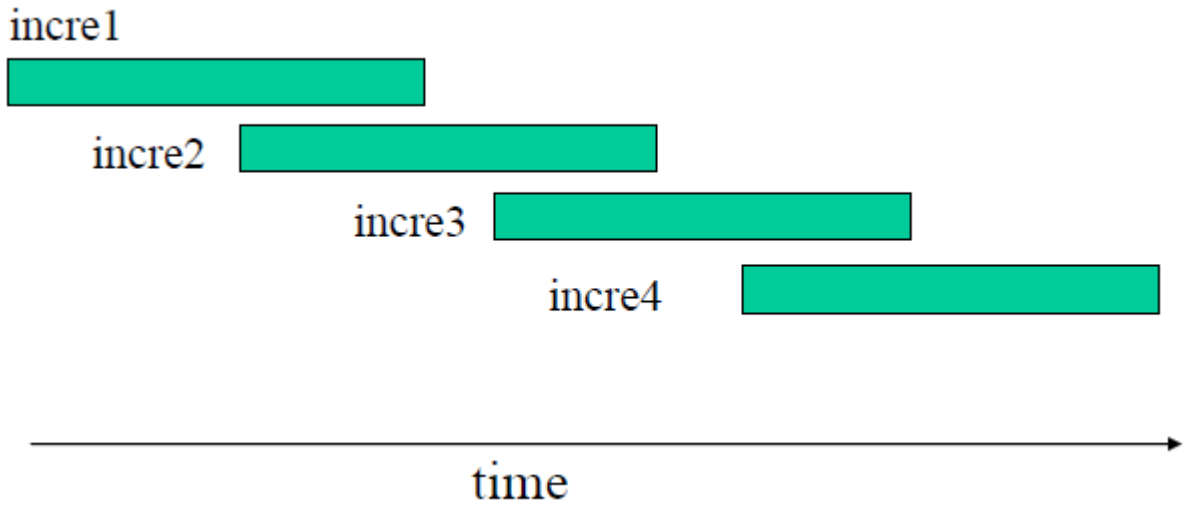
- Topt est le calendrier optimal calculé en utilisant la COCOMO-II équation d'horaire

COCOMO81: Le EM pour la compression Horaire

- Supposons que l'effort et calendrier prévisionnel pour un projet est de 10 personnes pendant 12 mois
  - ➔ Effort estimé = 10 x 12 = 120 mois-personne
- Maintenant, supposons que le calendrier est comprimé par 25% à 9 mois
  - ➔ T / topt = 0,75
- Effort accru = 120 x 1,43 = 172 mois-personne
- nécessaire du personnel # = 172 / 9 = 19 personnes

COCOMO-II:

- Estimation des entrées sont les suivants:
  - taille estimée de chaque incrément
  - les facteurs d'ajustement pour chaque incrément
  - temps relatif de début de chaque incrément
  - la fin estimée de chaque incrément



Le contenu est adapté de Fairley (2009)

COCOMO-II: Effort Estimation Equation

- Effort en mois-personnes (PM) est estimée en utilisant une équation de la forme:

  PM = 2.94 x (taille)^{B} * Π (17 multiplicateurs d'effort)

- • L'exposant B est de la forme:

  B = 0,91 + 0,01 x Σ (cinq facteurs d'échelle)

  Les facteurs d'échelle résumer à une valeur entre 0 et ~ 31,6

  B est donc dans la gamme de 0,91 à 1,23

- Les facteurs d'échelle exposant sont:
  - **PREC:** precedentedness (familiarité) du système
  - **FLEX:** flexibilité dans les exigences
  - **RESL:** degré de résolution de risque et la spécification d'interface au design review
  - **EQUIPE:** la cohésion de l'équipe de développement
  - **PMAT:** La maturité des processus (CMM)

COCOMO-II: Cost Drivers (inducteurs de coût)

Facteurs de produit

- RELY: la fiabilité requises
- DATA: rapport de la taille des données à la taille du code
- CPLX: la complexité du produit
- **RUSE**: l'effort nécessaire à la réutilisation *
- **DOCU:** documentation correspondent aux besoins du cycle de vie*

- Facteurs Platform
  - TIME: contrainte de temps d'exécution sur le processeur cible
  - STOR: contrainte de mémoire sur le processeur cible
  - PVOL: volatilité plateforme

*** Nouveau en COCOMO II**

COCOMO-II: Cost Drivers (inducteurs de coût)

Facteurs de personnel

- ACAP: la capacité d'analyste
- PCAP: la capacité de programmation
- APEX: expérience d'application
- PLEX: l'expérience plateforme
- LTEX: expérience dans les langages et outils
- **PCON:** * la continuité du personnel

Facteurs de projet

- Outil: L'utilisation d'outils logiciels
- **SITE**: sites de développement multiples *
- SCED: calendrier de développement nécessaires

*** Nouveau en COCOMO II**

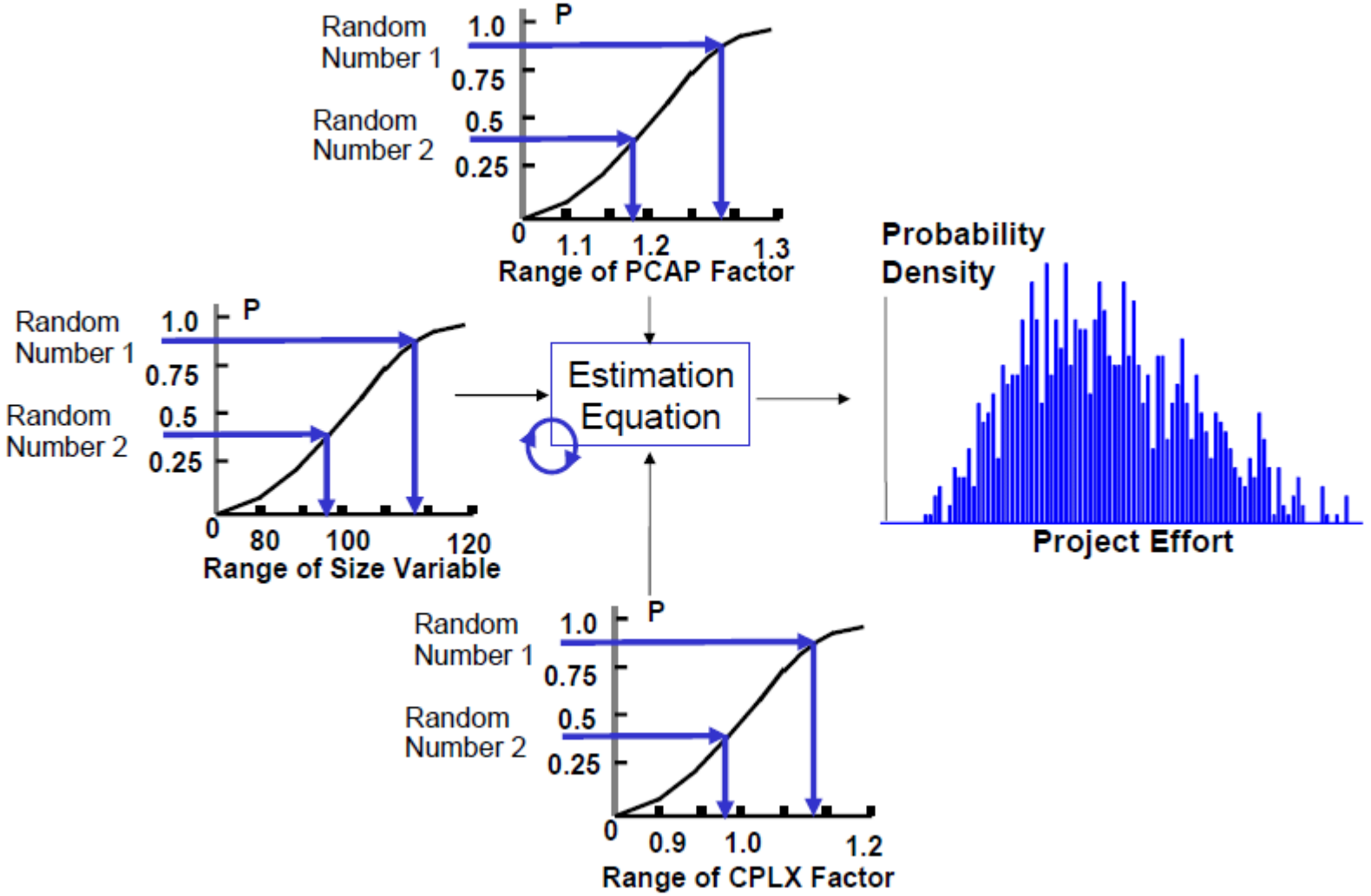
COCOMO-II: Estimations additionnels

- COCOMO fournit des estimations des:
  - horaire et l'effort total
  - effort et le calendrier par phase de développement
  - effort et calendrier de 7 types d'activités de travail au sein de chaque phase du développement
  - jalons mensuels, frais et dépens à ce jour
  - premières estimations (pre-architecture)
  - estimations raffinées (post-architecture)
- COCOMO-II supporte des estimations pour
  - développement progressif
  - réutilisation de composants logiciels
  - développement de composants pour réutilisation
  - points de fonction ou de lignes de code

PERT + Modèles d'estimation paramétrique

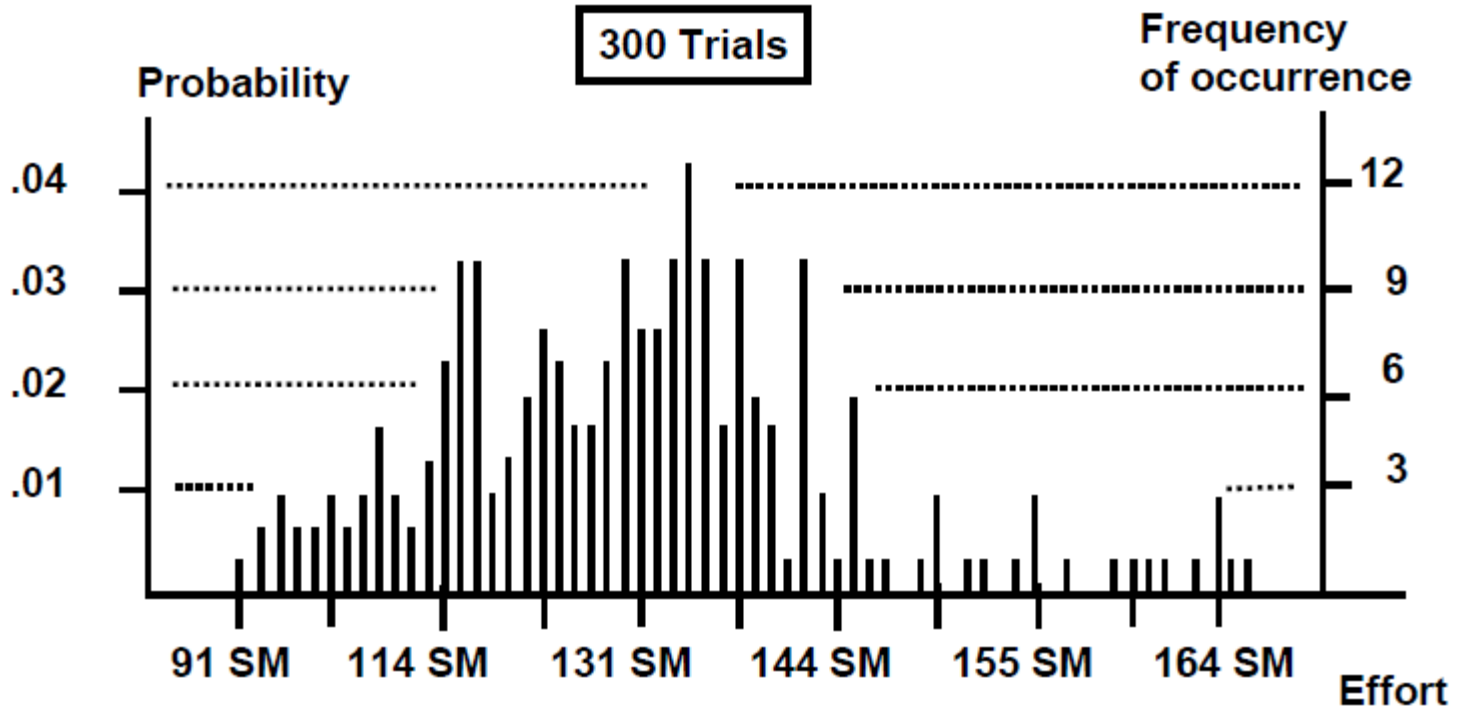
- Une approche PERT peut être utilisé avec SLIM ou COCOMO pour produire des estimations probabilistes.
- Trois estimations sont données pour la taille du produit et pour chacun des facteurs de coûts
  - a: la plus petite valeur probable (par exemple à 20% probables)
  - m: valeur la plus probable (par exemple, 50% * valeur probable)
  - b: plus grande valeur probable (par exemple, 80% * valeur probable)
- calculer l'effort, E, et l'écart-type, σ de taille et de chaque facteurs de coûts:
  - $E = (a + 4m + b) / 6$ & $\sigma = (b - a) / 3.2$
- Utilisez simulation Monte Carlo pour estimer la distribution de probabilité des efforts estimés du projet

PERT + Monte Carlo Estimation



Le contenu est adapté de Fairley (2009)

PERT + Monte Carlo Estimation



Le contenu est adapté de Fairley (2009)