

Avec les compliments de **IBM**

Édition limitée IBM

# Ingénierie Système

POUR  
**LES NULS**

## *Vous apprendrez :*

- Ce qu'est l'ingénierie système
- Comment l'ingénierie système peut vous aider à mettre au point des produits intelligents et connectés
- Comment accélérer les délais de mise sur le marché, garantir la souplesse de votre entreprise et fournir des produits intelligents de qualité
- Comment limiter les coûts



**Cathleen Shamieh**



***Ingénierie  
Système***  
POUR  
**LES NULS<sup>®</sup>**  
ÉDITION LIMITÉE IBM

**Cathleen Shamieh**



WILEY

Wiley Publishing, Inc.

## Ingénierie Système pour LES NULS®, édition limitée IBM

Publié par

**Wiley Publishing, Inc.**

111 River Street

Hoboken, NJ 07030-5774 États-Unis

[www.wiley.com](http://www.wiley.com)

Copyright © 2012 par Wiley Publishing, Inc., Indianapolis, Indiana États-Unis

Publié par Wiley Publishing, Inc., Indianapolis, Indiana États-Unis

Aucun extrait de cette publication ne peut être reproduit, stocké dans une base de données ni transmis, sous quelque forme ou par quelque moyen que ce soit (électronique, mécanique, photocopie, enregistrement, numérisation ou autre), sauf aux conditions autorisées aux alinéas 107 et 108 du United States Copyright Act de 1976, en l'absence d'autorisation écrite préalable de l'Éditeur. Les demandes d'autorisation doivent être adressées par courrier à Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030 États-Unis, par téléphone au +1 (201) 748 6011, par télécopie au +1 (201) 748 6008, ou en ligne sur <http://www.wiley.com/go/permissions>.

**Marques :** Wiley, le logo Wiley Publishing, Pour les Nuls, le logo Dummies Man, A Reference for the Rest of Us!, The Dummies Way, Dummies.com, Making Everything Easier, et les appellations commerciales afférentes sont des marques commerciales ou déposées de John Wiley & Sons, Inc. et/ou de ses sociétés affiliées aux États-Unis et dans d'autres pays, dont l'utilisation est interdite en l'absence d'autorisation écrite. Toutes les autres marques citées sont la propriété de leurs détenteurs respectifs. Wiley Publishing, Inc. n'est lié à aucun des produits ou fournisseurs cités dans cet ouvrage.

**LIMITATION DE RESPONSABILITÉ/DÉNI DE GARANTIE :** L'ÉDITEUR ET L'AUTEUR S'ABSTIENNENT DE TOUTE DÉCLARATION OU GARANTIE S'AGISSANT DE L'EXACTITUDE OU DE L'EXHAUSTIVITÉ DU CONTENU DE CET OUVRAGE, ET REJETTENT EN PARTICULIER TOUTE GARANTIE, Y COMPRIS, NON LIMITATIVEMENT, TOUTE GARANTIE D'APTITUDE À UN USAGE PARTICULIER. AUCUNE GARANTIE NE PEUT ÊTRE CONSENTIE OU ÉTENDUE AU TITRE D'UN DOCUMENT COMMERCIAL OU PROMOTIONNEL. LES CONSEILS ET STRATÉGIES PRÉSENTÉS ICI RISQUENT DE NE PAS CONVENIR À TOUTES LES SITUATIONS. CET OUVRAGE EST COMMERCIALISÉ, SACHANT QUE L'ÉDITEUR NE DISPENSE AUCUN SERVICE JURIDIQUE, COMPTABLE OU AUTRES SERVICES PROFESSIONNELS. SI UNE ASSISTANCE PROFESSIONNELLE EST REQUISE, LES SERVICES D'UN PROFESSIONNEL COMPÉTENT DEVRONT ÊTRE SOLlicitÉS. NI L'ÉDITEUR, NI L'AUTEUR NE POURRONT ÊTRE TENUS RESPONSABLES DES DOMMAGES DÉCOULANT DES PRÉSENTES. SI UN ÉTABLISSEMENT OU SITE WEB EST RÉFÉRENCÉ DANS UNE CITATION ET/OU COMME SOURCE POTENTIELLE D'INFORMATIONS COMPLÉMENTAIRES DANS CET OUVRAGE, CELA NE SIGNIFIE AUCUNEMENT QUE L'AUTEUR OU L'ÉDITEUR AVAILISE LES INFORMATIONS SUSCEPTIBLES D'ÊTRE COMMUNIQUÉES PAR CET ÉTABLISSEMENT OU CE SITE WEB OU SES RECOMMANDATIONS. PAR AILLEURS, LE LECTEUR DOIT AVOIR CONSCIENCE QUE LES SITES WEB CITÉS DANS CET OUVRAGE PEUVENT AVOIR ÉVOLUÉ OU DISPARU ENTRE LE MOMENT OÙ CE LIVRE A ÉTÉ ÉCRIT ET CELUI OÙ IL EST LU.

Pour toute information d'ordre général sur nos autres produits et services, veuillez contacter notre service de développement commercial aux États-Unis par téléphone au +1 317 572 3205. Pour obtenir plus de détails utiles sur la création d'un ouvrage *Pour LES NULS* personnalisé, adapté à votre entreprise, contactez [info@dummies.biz](mailto:info@dummies.biz). Pour de plus amples informations sur l'exploitation sous licence de la marque *Pour LES NULS* avec des produits ou services, contactez [BrandedRights&Licenses@Wiley.com](mailto:BrandedRights&Licenses@Wiley.com).

ISBN : 978-1-118-34839-0

Fabriqué aux États-Unis

10 9 8 7 6 5 4 3 2 1



## Note de l'éditeur

Nous sommes fiers de ce livre et des personnes qui ont travaillé dessus. Pour obtenir tous détails utiles sur la création d'un ouvrage *Pour LES NULS* personnalisé, adapté à votre entreprise, contactez [info@dummies.biz](mailto:info@dummies.biz). Pour tous détails sur l'exploitation sous licence de la marque *Pour LES NULS* avec des produits ou services, contactez [BrandedRights&Licenses@Wiley.com](mailto:BrandedRights&Licenses@Wiley.com).

Certaines personnes ont contribué à la mise sur le marché de cet ouvrage, notamment :

### **Achats, éditorial et développement des médias**

**Éditrice de projet** : Carrie A. Burchfield

**Responsable éditorial** : Rev Mengle

**Éditrice sénior pour les achats** :  
Katie Feltman

**Représentante du développement  
commercial** : Sue Blessing

**Spécialiste du projet d'édition** :  
Michael Sullivan

### **Services de composition**

**Coordinatrice de projet sénior** : Kristie Rees

**Mise en page et graphisme** : Melanee Habig

**Relectrice** : Jessica Kramer

---

### **Publication et éditorial pour les débutants en technologies**

**Richard Swadley**, Vice-président et éditeur en chef du groupe

**Andy Cummings**, Vice-président et éditeur

**Mary Bednarek**, Directrice exécutive, Achats

**Mary C. Corder**, Directrice éditoriale

### **Publication et éditorial pour les débutants en produits de consommation**

**Diane Graves Steele**, Vice-présidente et éditrice, série pour les consommateurs

### **Services de composition**

**Debbie Stailey**, Directrice des services de composition

### **Développement commercial**

**Lisa Coleman**, Directrice, Nouveaux marchés et développement de la marque

# Aperçu du contenu

---

<b>Introduction</b> .....	<b>1</b>
<b>Chapitre 1 : Créer des produits intelligents</b> .....	<b>3</b>
<b>Chapitre 2 : Dompter le fauve avec Ingénierie Système</b> .....	<b>13</b>
<b>Chapitre 3 : Révolution des exigences</b> .....	<b>23</b>
<b>Chapitre 4 : Abstraire le système par la modélisation</b> .....	<b>35</b>
<b>Chapitre 5 : Garantir une excellente qualité</b> .....	<b>43</b>
<b>Chapitre 6 : Aider les grandes équipes à collaborer et à gérer le changement</b> .....	<b>51</b>
<b>Chapitre 7 : Dix manières de réussir son Ingénierie Système</b> .....	<b>61</b>

# Introduction



**L**es produits intelligents sont partout. Ils contrôlent l'éclairage des routes, pilotent des avions et vous aident à atteindre votre destination. Ils sont au cœur des systèmes et des services que vous utilisez au quotidien : des smartphones aux voitures intelligentes en passant par les systèmes médicaux, l'aérospatiale et la défense.

Ces produits intelligents et interconnectés révolutionnent la manière dont nous interagissons et réalisons nos tâches quotidiennes. En combinant électronique, logiciels, capteurs et divers types de matériel, nous disposons de la technologie pour créer des produits multifonctionnels personnalisés. Et grâce à une imagination fertile nous pouvons créer d'innombrables systèmes et services novateurs et personnalisés apportant une véritable valeur ajoutée.

Le principal défi lorsqu'il s'agit de créer des produits intelligents réside dans la manière de s'organiser : comment peut-on intégrer efficacement des technologies complexes pour créer un « système de systèmes » intelligent ? Lequel devra bien sûr tenir ses promesses et exprimer tout son potentiel. L'ingénierie système est la solution.

*Ingénierie Système pour les Nuls*, édition limitée IBM, explique ce qu'est l'ingénierie système et comment elle peut vous aider à mettre au point plus facilement des produits intelligents et connectés. Si vous cherchez à accélérer les délais de mise sur le marché, rendre votre entreprise plus agile et proposer des produits intelligents de qualité tout en limitant vos dépenses, *Ingénierie Système pour les Nuls*, est le livre qu'il vous faut.

## *Comment ce livre est organisé*

Les sept chapitres qui composent ce livre sont conçus pour vous aider à comprendre les problèmes que cherche à résoudre l'ingénierie système et détaillent les étapes de sa mise en œuvre. Voici un aperçu du contenu :

- ✔ Le chapitre 1 explique ce que sont les produits intelligents et pourquoi ils demandent une nouvelle approche de développement.
- ✔ Le chapitre 2 offre une présentation de haut niveau de l'ingénierie système.
- ✔ Le chapitre 3 définit le rôle essentiel des exigences tout au long du cycle de développement des systèmes.
- ✔ Le chapitre 4 montre comment les modèles peuvent renforcer votre compréhension de la structure des systèmes et de leur comportement.
- ✔ Le chapitre 5 explique comment faire pour concevoir un système sans défaut (vérification) et répondant aux besoins (validation).
- ✔ Le chapitre 6 propose des moyens d'améliorer la collaboration entre les équipes de développement.
- ✔ Le chapitre 7 rappelle les dix règles fondamentales pour une ingénierie des systèmes réussie.

## Les icônes utilisées dans ce livre

Ce livre utilise les icônes suivantes pour attirer votre attention sur les informations que vous pouvez trouver utiles.



Cette icône vous signale les concepts importants à retenir.



Cette icône indique les suggestions qui vous faciliteront la vie.



Cette fois, il ne s'agit plus de conseils. Les paragraphes signalés par cette icône attirent votre attention sur les pièges les plus courants dans lesquels vous risquez de tomber.



Cette icône signale les endroits où des sujets techniques sont évoqués en détail. Vous pouvez donc les ignorer si vous voulez.



# Chapitre 1

## Créer des produits intelligents

### *Dans ce chapitre*

- ▶ Faire face à la demande de systèmes intelligents et interconnectés
- ▶ Reconnaître les défis à relever pour une ingénierie réussie
- ▶ Passer à la vitesse supérieure pour élargir son champ d'action

Imaginez : tandis que vous sortez du garage, la voiture envoie un signal à votre maison pour armer le système d'alarme et fermer la porte du garage. Votre téléphone se synchronise automatiquement avec le système de commande vocale de votre voiture et le système de navigation GPS analyse le trafic en direct pour vous proposer le meilleur itinéraire pour vous rendre au travail. Elle passe en revue votre agenda sur votre smartphone et vous propose différents rendez-vous. Votre voiture vous signale qu'il est temps de penser à changer l'huile et vous demande si vous souhaitez appeler dès maintenant votre garagiste pour fixer rendez-vous. Votre voiture peut même vous signaler les conditions de trafic attendues pour le retour.

Est-il possible qu'un véhicule qui n'était jadis qu'une vulgaire boîte de métal soit devenu si incroyablement intelligent et utile ? Absolument !

Dans ce chapitre, vous découvrirez ce qui fait fonctionner ces produits intelligents, comment ils interagissent et pourquoi vous devez opter pour de nouveaux procédés de développement.

## Qu'est-ce qui rend ces produits si intelligents ?

Les produits intelligents font fureur de nos jours. Difficile d'imaginer la vie avant les appareils ménagers programmables, les jeux vidéo interactifs et les téléphones mobiles polyvalents capables de prendre des photos ou des vidéos, surfer sur Internet et lire de la musique. Les avions se succèdent tout en évitant les collisions tandis que l'on fait appel aux drones intelligents et autres systèmes de défense de précision pour assurer notre sécurité.

Mais qu'est-ce qui permet à de tels objets inanimés d'accomplir des faits si étonnants ?

### Apporter une valeur ajoutée grâce aux systèmes intelligents

Les systèmes intelligents pilotés par logiciel apparaissent un peu partout :

- ✓ **Santé** : des logiciels spécifiques offrent un accès fiable et sécurisé à des images médicales complexes et aux rapports à partir d'appareils mobiles. Les médecins peuvent ainsi passer en revue les données du patient et établir rapidement un diagnostic.
- ✓ **Services publics** : une communication bidirectionnelle entre les fournisseurs d'énergie et les consommateurs par le biais d'un « réseau intelligent » permet un contrôle intelligent de la consommation d'énergie. Par exemple, les lave-linge peuvent être activés par le réseau lorsque l'électricité coûte le moins cher. De même, certains appareils peuvent être désactivés pendant les heures de pointe. Les voitures intelligentes constituent un nouvel exemple.
- ✓ **Appareils intelligents** : les appareils ménagers connectés offrent un aperçu de l'état de la consommation d'énergie grâce à des interfaces intuitives. Le contrôle à distance de ces appareils permet aux consommateurs d'atteindre le niveau de confort souhaité tout en limitant leur consommation d'énergie.
- ✓ **Divertissement** : les téléviseurs intelligents offrent un accès à l'Internet sans fil. Les consommateurs peuvent ainsi louer des films, faire leurs achats, consulter les bulletins météo, afficher des pages d'information et télécharger des applications.
- ✓ **Automobile** : les systèmes intelligents d'évitement de collision intègrent la technologie GPS différentielle, la communication sans fil et l'affichage graphique dans le véhicule afin d'alerter les conducteurs en présence de véhicules en approche. Ils pourront même effectuer les manœuvres d'évitement d'urgence.

## Mélanger les composants technologiques

Aujourd'hui, les produits et les services intelligents sont le fruit de la rencontre des technologies de fabrication, de l'information et de l'électronique. Certains fabricants innovants ont réalisé qu'ils pouvaient tirer profit des avancées considérables faites sur les composants microélectroniques, logiciels, mécaniques, incluant capteurs et actionneurs, pour créer des produits capables de séduire leurs clients et de devancer leurs concurrents. Ils ont pris de-ci de-là, ont mélangé le tout avec l'aide de leurs amis ingénieurs et le tour était joué ! Ils disposaient de produits qui auraient pu étonner Géo Trouvetou lui-même.



Les produits intelligents se présentent sous différentes formes, mais en général, ils regroupent ces trois caractéristiques :

- ✔ **Équipés** : les produits intelligents intègrent une grande variété de composants comme des caméras, détecteurs de mouvement, récepteurs sans fil, capteurs de position, de son, de chaleur, d'humidité et de champs magnétiques. Ils parviennent ainsi à surveiller en permanence leur état et obtenir des informations sur ce qui les entourent. Ainsi, les produits intelligents parviennent à s'adapter en temps réel à leur environnement.
- ✔ **Interconnectés** : lorsque deux produits ou plus interagissent et partagent des données, ils offrent une valeur conjointe supérieure à celle de chaque produit isolé. Reliez-les à Internet, à un système d'administration ou d'information et ils n'auront plus de limites !
- ✔ **Intelligents** : en exploitant des informations obtenues de capteurs, des historiques et des données des utilisateurs, les produits bien conçus parviennent à faire des prévisions, améliorer les résultats et faciliter la vie des utilisateurs.

## Les intégrer tous par du logiciel

La révolution des produits intelligents ne dépend pas seulement de l'augmentation phénoménale de la puissance de calcul. Bien sûr, les microprocesseurs intégrés dans les produits offrent la puissance de traitement. Mais c'est le logiciel embarqué qui fournit la fonctionnalité attendue en exécutant les algorithmes, analysant les gros volumes de données et contrôlant tous les composants électroniques ou mécaniques du produit.

Pour accomplir la tâche qui leur est confiée, les microprocesseurs vont exécuter des milliers, voire des millions, de lignes de code. Par exemple, les voitures de luxe modernes contiennent des dizaines de microprocesseurs qui exécutent jusqu'à 100 millions de lignes de code afin de proposer entre 250 à 300 fonctions au conducteur et à ses passagers.

Tandis que les fabricants de semi-conducteurs mettent au point des microprocesseurs de plus en plus puissants, les développeurs peuvent désormais repousser les limites de leur créativité. Et c'est une bonne chose étant donné que la plupart des composants matériels qui différencient les produits se standardisent de plus en plus. Très souvent, c'est le logiciel et non le matériel qui permet à un produit de se distinguer et de gagner des parts de marché.



La polyvalence inhérente aux logiciels offre la possibilité de développer des nouvelles fonctionnalités en grand nombre, permettant ainsi aux fabricants d'adapter leurs produits et de répondre aux besoins des clients en particulier en terme d'innovation. Par exemple, les lecteurs MP3 les plus en vogue actuellement ne font pas que lire la musique. Ils permettent de gérer une bibliothèque complète, lire la vidéo en direct, envoyer et recevoir des messages, jouer à des jeux vidéo et exécuter des applications tierces. Les meilleurs produits peuvent même être actualisés afin de s'adapter aux nouvelles exigences du marché.

## *Favoriser la diffusion grâce aux normes ouvertes*



D'ici 2013, il y aura 1,2 milliard d'appareils connectés répartis sur 800 millions de foyers dotés d'un accès à haut débit. Ces chiffres donneraient presque le tournis. Si vous avez de bonnes idées et comptez proposer par Internet des services offrant de la valeur ajoutée à vos clients, ces chiffres vous font probablement saliver.

Imaginez si tous les appareils électroniques étaient créés à partir de rien et si chaque fabricant avait sa propre manière de se connecter à Internet pour communiquer avec le monde extérieur. Il pourrait y avoir du monde sur le réseau, mais tout cela serait inutile. Ce serait vraiment passer à côté de quelque chose !

Ne vous inquiétez pas ! Certains petits malins ont imaginé l'avenir et pensé à créer des *normes ouvertes*. Il s'agit en fait de méthodes communes utilisées pour envoyer des messages entre les appareils et les fournisseurs de service.

Plus les entreprises opteront pour des protocoles de communication basés sur des normes ouvertes, plus nombreux seront les produits et services offerts à chacun ; la population évolue rapidement vers « l'Internet des choses », cet écosystème composé de produits et services intelligents et connectés.

## Répondre aux besoins des clients

Une grande partie de la population maîtrise déjà les produits intelligents.



Rien qu'en 2010, quelque 40 millions d'appareils de navigation ont été vendus dans le monde entier. En 2008, plus de 55 millions de personnes se sont attachés des iPod.

Si vous voyagez souvent, vous n'avez-vous peut-être aucune inquiétude à prendre l'avion car vous savez qu'il est équipé d'instruments intelligents assurant votre sécurité. Les utilisateurs avertis sont conscients des atouts qu'offrent les produits intelligents modernes et certains sont même demandeurs de versions encore plus avancées.

Aujourd'hui, les consommateurs exigent des produits fiables, interactifs et performants et ils les veulent dès maintenant. Si vous êtes suffisamment chanceux (et malin) pour produire à temps, ne vous arrêtez pas là, car avant que vous n'avez le temps de dire « ouf », vos clients attendront déjà une nouvelle version.



En associant des fonctions de personnalisation et d'intégration à vos modèles, vous créez des produits qui répondront aux besoins individuels de chaque utilisateur en s'adaptant à son environnement. Les clients cherchent des produits qui leur faciliteront la vie et la leur rendront plus agréable. Chacun a sa propre manière d'utiliser un produit, ils en attendent certaines fonctions et un certain comportement. Les clients exigent aujourd'hui des produits intelligents aussi faciles à personnaliser que de télécharger la dernière application pour smartphone !

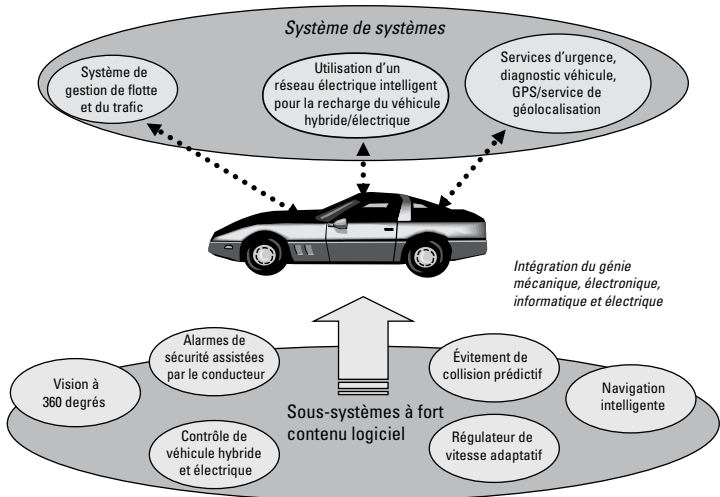
## Réaliser que les produits intelligents ne sont pas isolés

Vous verrez qu'il ne faut pas toujours très longtemps pour créer un produit fonctionnel, à la mode et autonome. Mais les produits d'aujourd'hui doivent en plus être suffisamment intelligents pour interagir entre eux.

Prenons l'exemple d'une voiture récente (v. Figure 1-1). Une voiture de luxe renferme typiquement un grand nombre de sous-systèmes à fort contenu logiciel destinés à rendre la conduite plus agréable tout en renforçant la sécurité et en réduisant la consommation : système de freinage ABS, système d'évitement de collision, commandes de confort, sécurité, etc. La conception et la fabrication d'une voiture intelligente, qui est véritablement un « système de systèmes », impliquent l'intégration complexe d'éléments mécaniques, électriques et électroniques, sans oublier le logiciel pouvant faire une bonne centaine de millions de lignes de code.

Pour compliquer un peu plus les choses, les voitures intelligentes interagissent aussi avec d'autres systèmes extérieurs à la voiture. Les services de positionnement, les systèmes de diagnostic véhicule et les systèmes de charge des modèles hybrides ou électriques ne sont que quelques-uns des nombreux systèmes avec lesquels une voiture peut entrer en interaction au sein d'un écosystème automobile plus vaste.

Cela vous semble compliqué ? Ça l'est ! Mais c'est aussi incroyablement utile. Par exemple, si le système de sécurité de votre voiture est conçu pour interagir avec les services d'urgence, il pourra transmettre des informations sur l'accident aux équipes d'intervention en fonction des données obtenues des capteurs embarqués. Des éléments critiques tels que la puissance du choc permettront aux secours de savoir quelles équipes envoyer sur place.



**Figure 1-1** : une voiture intelligente est un système de systèmes évoluant dans un écosystème plus vaste.



Souvent, les fonctions les plus importantes d'un produit intelligent ne se trouvent pas toutes dedans. Elles peuvent s'enclencher suite aux interactions avec d'autres produits ou services de l'écosystème. À l'instar des personnes, les produits intelligents doivent collaborer et partager les informations !



Ce n'est pas parce que vous *pouvez* partager les données que vous *devez* le faire ! La confidentialité, la sécurité et les réglementations notamment risquent d'avoir une influence majeure sur votre conception !

## Relever de nouveaux défis

La mise au point d'un produit intelligent à même d'offrir un service personnalisé à un grand nombre de clients exigeants et pressés n'est pas chose facile. Vous êtes peut-être un expert dans votre discipline, mais vous risquez de vite constater que ni votre formation poussée ni votre longue expérience ne vous ont préparé à l'énorme défi auquel vous devez faire face aujourd'hui.



Avant de vous lancer, vous devez tenir compte des éléments suivants pour mettre au point des produits intelligents :

- ✔ **Maîtriser plusieurs compétences** : vous devez être expert dans différents domaines techniques, dont la fabrication, l'électronique, le génie mécanique et le développement de logiciels. Bien que la plupart des entreprises se distinguent dans l'une ou l'autre de ces disciplines, l'on ne les trouve toutes que très rarement sous le même toit.
- ✔ **Devenir une société de services et d'ingénierie informatique de niveau international** : si vous fabriquez des produits et que vous considérez que les logiciels sont un mal nécessaire, mieux vaut sans doute passer sous hypnose, car l'essentiel de l'intelligence des produits dits intelligents provient de ces logiciels. Malheureusement, le codage ne se fait pas tout seul... Du moins, pas encore.
- ✔ **Intégrer les activités de développement matériel et logiciel** : bien que les fonctions pilotées par logiciel occupent le centre du tableau, vous devez faire en sorte que vos équipes responsables du matériel et des logiciels collaborent *réellement*. Elles ne doivent en aucun cas se limiter à se passer entre deux portes les modules développés en vue de l'intégration et des essais.
- ✔ **Gérer efficacement des équipes distribuées** : si vos équipes de développement sont réparties sur plusieurs villes, plusieurs pays et/ou plusieurs fuseaux horaires, essayez de favoriser la synergie pour garantir des résultats efficaces, précis et collaboratifs.

- ✔ **Permettre les interactions avec les autres systèmes** : vous ne voulez sans doute pas être le seul produit en lice incapable de se connecter à Internet, aux systèmes informatiques et administratifs, ou tout autre système interconnecté. Les produits autonomes ne sont pas si intelligents que ça. Ils sont d'ailleurs enclins à se muer en échecs isolés.
- ✔ **Garantir la conformité** : même si votre produit n'est pas directement lié à des normes légales ou industrielles, il pourra avoir à interagir avec un produit ou un service qui l'est. Mieux vaut donc jouer la sécurité pour faire en sorte que tout le monde veuille jouer avec vous.
- ✔ **Faire évoluer vos priorités en matière de conception** : privilégiez une conception évolutive plutôt qu'un produit figé. Il est essentiel d'avoir les priorités *adéquates* en termes de conception. Dans l'aéronautique, il s'agira de la sécurité des passagers. La défense quant à elle privilégiera la sécurité du pays tandis que l'électronique favorisera la mise à niveau ultérieure.
- ✔ **Réduire les cycles de vie des produits** : tandis que la demande de nouvelles fonctions réduit la durée de service des meilleurs produits, vous devez faire en sorte de toucher le marché au bon moment en lui proposant les caractéristiques adéquates.
- ✔ **S'adapter à des exigences en perpétuelle évolution** : les besoins du client qui évoluent, un marché dynamique où la concurrence fait rage, de nouveaux objectifs d'entreprise sont autant de raisons valables pour des changements qui peuvent s'avérer irritants. Si vous êtes malin, vous saurez comment tirer profit de ces changements.

## Sonder le terrain

Si vous ne parvenez pas à relever les défis associés à la mise en place d'un nouveau produit intelligent, vous risquez la douche froide - de ternir la réputation de votre entreprise et de faire chuter votre chiffre d'affaires.

Les petites erreurs faciles à résoudre avec des produits autonomes sont souvent multipliées et se répercutent sur toute la chaîne dans le cas des systèmes interconnectés. Une panne logicielle peut annihiler des mois d'effort si vous ne le résolvez pas suffisamment tôt lors du développement. Vous risquez ainsi d'augmenter vos coûts et prendre du retard. Comme le volume de logiciel intégré aux appareils double environ tous les deux ans, on comprend aisément pourquoi 66 % des composants logiciels dépassent leur budget initial et 24 % des gros projets sont annulés en raison des retards sur le planning.





Si vous ne parvenez pas à mettre au point des produits complexes au cours de cycles plus courts tout en conservant la même qualité, vous risquez de perdre de l'argent et d'entacher la réputation de votre marque. Pourtant, les produits intelligents et interconnectés sont souvent associés à des centaines, voire des milliers d'exigences. Difficile donc d'imaginer maintenir la qualité tout en réduisant les cycles de développement.

Si vous n'êtes pas capable de répondre rapidement aux nouvelles demandes du marché ni parer les attaques de la concurrence, ne soyez pas surpris si vous perdez des parts de marché pour le plus grand bonheur des entreprises plus réactives. Pour de nombreuses sociétés, réaliser le projet initial est déjà suffisamment ardu. Presque un tiers des produits fabriqués aujourd'hui ne respectent pas les critères de performance ou de fonctionnement. On peut donc s'attendre à ce que ces appareils soient bien vite mis sur la touche !

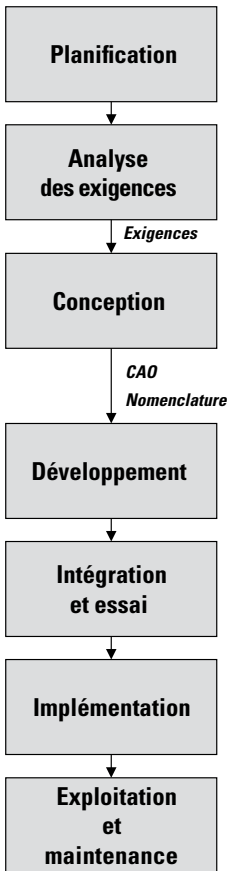


Ce n'est pas parce que vous maîtrisez la technique et travaillez consciencieusement sur la conception du système que vous réussirez. Bien souvent les projets échouent en raison de problèmes au niveau des caractéristiques techniques de l'interface d'un sous-système, du non-respect d'exigences ou de lacunes en termes de transfert des connaissances stratégiques. Ce n'est donc pas l'ingénierie qui fait défaut ici.

## *Prendre conscience du besoin d'un changement d'optique*

Vous devez accepter que, pour réussir sur le marché actuel, vous devez changer votre manière d'apporter de la valeur ajoutée par vos produits. Lorsque vous aurez compris cela, vous pourrez reconsidérer les processus de planification, de gestion et de développement des produits. Vous verrez d'emblée le besoin de passer d'un intérêt pour les coûts à l'intérêt pour l'innovation et le changement, tout en considérant le logiciel comme l'élément fort de différenciation.

Par le passé, à l'époque où le matériel régnait en maître, la conception 3D assistée par ordinateur (CAO) et la gestion de la nomenclature faisaient office de pierres angulaires d'un développement séquentiel de produit (v. Figure 1-2). Votre équipe de développement a utilisé un système de CAO pour concevoir le matériel de sorte qu'il réponde à une série de critères. L'équipe informatique a travaillé quant à elle sur une série d'exigences différentes, quoique connexes, afin de développer le code nécessaire. Votre modèle CAO et la nomenclature ont été transmis à l'atelier, qui a défini la méthode de production la plus rapide et la moins chère. Enfin, les ingénieurs responsables de l'intégration et des essais opérant dans un service distinct ont chargé le logiciel et testé le produit.



**Figure 1-2 :** séquence de développement d'un produit.

À l'heure actuelle, ce type de méthode de développement séquentiel appuyé sur des documents ne suffit plus. Imaginez réagir rapidement à des événements imprévus sur le marché ou à de nouvelles exigences. Chaque changement demande que vous recommenciez depuis le début et passiez par chaque étape du processus. Votre entreprise pourrait ne pas s'en remettre.



Si l'avenir de votre entreprise repose sur le développement d'un nouveau produit intelligent, vous devez vous débarrasser de ces processus séquentiels et basés documents, désormais obsolètes, pour vous tourner vers une nouvelle approche qui vous ouvrira les portes du succès.

## Chapitre 2

# Dompter le fauve avec Ingénierie Système

### *Dans ce chapitre*

- ▶ Évaluation de la portée de l'ingénierie système
- ▶ Tempérer l'euphorie du projet pour faire face à la réalité
- ▶ Faire un modèle de conception

**U**ous n'êtes-vous jamais demandé comment la NASA parvenait à mettre au point des systèmes aussi complexes que les missions Apollo ? Comment des équipes éclectiques composées d'ingénieurs, de programmeurs, d'équipementiers et autres sont-elles parvenues à travailler ensemble pour permettre à l'homme de marcher sur la lune ? Cela n'aura sans doute pas été possible sans le recours à l'ingénierie système.

*L'ingénierie système* est une approche interdisciplinaire permettant de créer des systèmes aussi vastes que complexes répondant à un ensemble précis d'exigences commerciales et techniques. L'aérospatiale et la défense utilisent l'ingénierie système depuis très longtemps. D'ailleurs, la plupart des enseignements qu'elles en retirés sont désormais appliqués dans d'autres secteurs. À une époque où les systèmes de transport, les réseaux électriques, les réseaux téléphoniques et informatiques deviennent de plus en plus intelligents, vous avez forcément besoin de méthodes d'avant-garde pour les mettre au point.



L'ingénierie système existe depuis les années 1940, mais elle n'est exploitée par la NASA et consorts que depuis les années 1990. C'est à ce moment-là que les fabricants ont commencé à transformer les produits ordinaires en systèmes intelligents, grâce au recours aux technologies de l'information. Le développement de produit entrain dans une nouvelle ère dans laquelle les logiciels occupent désormais le rôle principal.

Puisqu'ils permettent de repousser les limites des fonctions des produits, les logiciels ont pris du galon dans différents domaines, autorisant ainsi de nouvelles interconnexions entre composants et établissant de nouveaux liens entre le système et son environnement. Plus de connexions impliquent un système encore plus complexe. L'entreprise n'a donc d'autre choix que d'abandonner le développement cloisonné pour privilégier des nouvelles méthodes collaboratives.

Dans ce chapitre, vous découvrirez ce qu'est l'ingénierie système, comment elle peut vous aider à gérer la complexité, à mettre au point des produits intelligents et à innover pour atteindre le succès.

## Apprendre à connaître l'ingénierie système

Si vous deviez demander à cinq experts ce que représente exactement l'ingénierie système, vous obtiendrez probablement cinq, voire six, réponses différentes. C'est notamment parce que les termes *ingénierie système* sont utilisés pour désigner plusieurs choses. Par ailleurs, le concept de système a évolué au fil des décennies. Les termes *ingénierie système* ne peuvent qu'évoluer eux aussi.

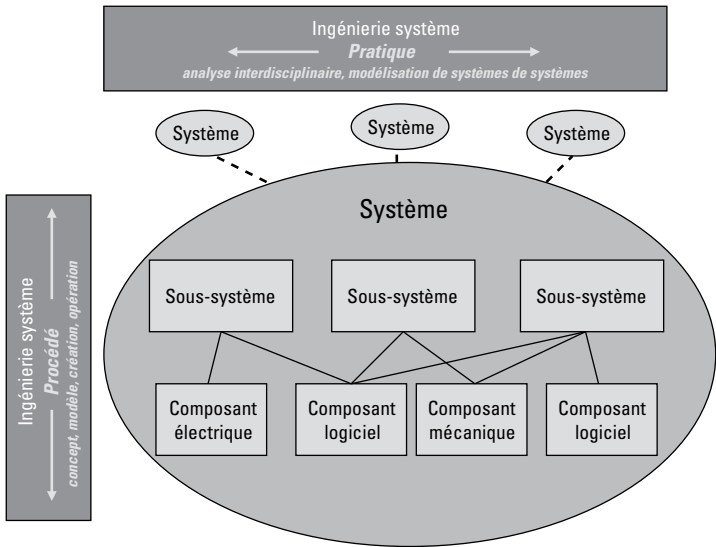
Mais ne vous inquiétez pas ! Bien que certains universitaires s'échineront probablement à vous expliquer les moindres détails et la portée de l'ingénierie système, la plupart des experts sont d'accord sur les bases. Dans ce chapitre, nous verrons quels sont ces éléments et comment les exploiter.

## Voir la forêt et les arbres



L'ingénierie système est à la fois une pratique et un processus (v. Figure 2-1) :

- ✓ En tant que pratique, elle privilégie la vue d'ensemble : les fonctions et le comportement global d'un système, les interfaces avec les utilisateurs et les autres systèmes, l'interaction de ses sous-systèmes et la manière d'associer les différentes disciplines de sorte à garantir leur synergie.
- ✓ En tant que processus, elle se veut une approche robuste et structurée du développement des systèmes, qui peut être appliquée au niveau d'un système de systèmes ou dans certaines disciplines d'ingénierie spécifiques.



**Figure 2-1 :** l'ingénierie système est à la fois une pratique et un processus.

Quelle que soit la manière dont vous l'analysez, l'ingénierie système consiste toujours à appliquer une certaine discipline au processus de développement. Et cette discipline se présente sous deux formes distinctes :

- **La discipline technique** veille à ce que vous mettiez en place un processus de développement rigoureux, de la conception à l'exploitation en passant par la production.
- **La discipline de gestion** organise les efforts techniques tout au long du cycle de vie du système en favorisant la collaboration, en définissant les méthodes de travail et en déployant les outils de développement.

## Suivre des principes directeurs

En associant l'ampleur et la profondeur, l'ingénierie système vous aide à contrôler avec précision un effort de développement complexe sans jamais perdre de vue les objectifs finaux du projet.

Commencez par respecter quelques principes de base :

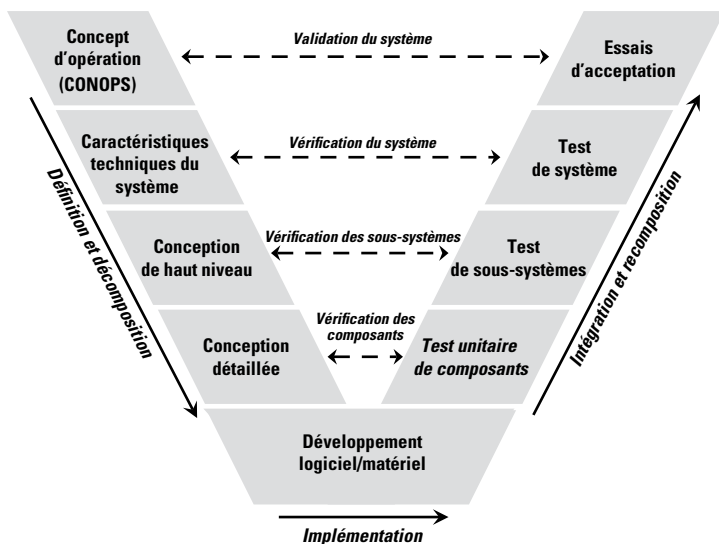


- ✔ **Ne perdez pas l'objectif de vue.** Définissez d'emblée les résultats visés et ne vous écartez en aucun cas de cet objectif, quelles que soient les circonstances.
- ✔ **Impliquez des partenaires stratégiques.** Obtenez l'avis des clients, utilisateurs, opérateurs, cadres et autres, quand vous prenez les décisions importantes aux différentes étapes du processus de développement.
- ✔ **Définissez le problème avant d'émettre l'hypothèse d'une solution.** En gardant l'esprit ouvert quant aux moyens permettant d'arriver à vos fins, vous pourrez trouver plusieurs alternatives et choisir la solution qui convient le mieux au résultat visé.
- ✔ **Divisez le problème en plusieurs parties faciles à gérer.** Décomposez votre système en systèmes plus petits. Divisez ensuite chaque sous-système en composants matériels et logiciels. Autre principe lié : gérer les interfaces entre les différentes parties afin d'en assurer l'intégration et pouvoir à terme déployer les compétences nécessaires.
- ✔ **Retardez les choix concernant les technologies spécifiques.** Attendez d'être bien avancé dans le projet avant de sélectionner les composants physiques. Vous évitez ainsi de vous imposer une technologie qui pourrait s'avérer obsolète ou superflue au moment où vous implémenterez votre conception.
- ✔ **Établissez le lien entre les exigences et la conception.** Veillez à pouvoir justifier vos décisions en matière de conception en les associant à des exigences techniques ou commerciales.
- ✔ **Testez tôt, testez souvent.** Profitez des avantages offerts par les prototypes, les simulateurs, les émulateurs et autres méthodes permettant à chacun d'évaluer le système suffisamment tôt. Veillez à ce que les tests montrent bien que vous répondez aux exigences en établissant des liens de traçabilité.

## Explorer le processus d'ingénierie système

Il est bon d'adopter des principes directeurs rigoureux, mais comment les mettre en pratique ? La première méthode consiste à mettre au point un processus homogène pour l'ingénierie système, capable d'englober ces principes.

Au cours de ces 20 dernières années, les experts des systèmes complexes ont mis au point et affiné le *modèle en V* (v. Figure 2-2). Ce modèle en V est une représentation graphique d'une série d'étapes et procédures permettant de développer des systèmes complexes.



**Figure 2-2 :** le modèle en V pour le processus d'ingénierie système.

En suivant le V de gauche à droite, vous exécuterez les différentes étapes du processus d'ingénierie système :

- **Concept d'opération (CONOPS) :** identifier et documenter les principaux besoins de vos parties prenantes, les capacités globales du système, les rôles et les responsabilités ainsi que les mesures de performances pour la validation du système en fin de projet.
- **Caractéristiques techniques du système :** dresser une liste décrivant les caractéristiques techniques vérifiables et répondant aux besoins des parties prenantes définies pendant l'étape CONOPS.
- **Conception de haut niveau :** concevoir une architecture de système de haut niveau répondant aux caractéristiques techniques et permettant la maintenance, les mises à jour et l'intégration à d'autres systèmes.

- ✓ **Conception détaillée** : détailler la conception jusqu'au niveau des composants et allouer les exigences à ce niveau. Les composants seront identifiés de manière à permettre leur développement ou leur achat dans le respect du budget initial.
- ✓ **Développement logiciel/matériel** : choisir la technologie adéquate. Développer le matériel et le logiciel identifiés en conception détaillée.
- ✓ **Test unitaire/de dispositifs** : tester les composants unitairement et vérifier leur bon fonctionnement sur la base des exigences allouées en conception détaillée..
- ✓ **Test de sous-systèmes** : intégrer les composants logiciels et matériels pour en faire des sous-systèmes. Tester et vérifier chaque sous-système sur la base des exigences de haut niveau.
- ✓ **Test de système** : intégrer les sous-systèmes et tester le système complet sur la base des exigences. Vérifiez que toutes les interfaces ont été correctement implémentées et que toutes les exigences et contraintes ont été respectées.
- ✓ **Essais d'acceptation** : vérifier que le système respecte les exigences et atteint les objectifs fixés.

Tout au long du processus d'ingénierie système, vous créez et affinez la documentation système. À chaque étape située sur la branche descendante du V dans la Figure 2-2, vous devrez définir les exigences permettant de passer à l'étape suivante ainsi que les jeux de test pour vérifier leur correcte implémentation à ce niveau. Par exemple, pendant l'étape « CONOPS », vous créez le document des exigences du niveau système qui guidera l'étape « Caractéristiques techniques du système », ainsi que le plan de validation du système qui guidera l'étape « Essais d'acceptation ». Pour chaque étape occupant la partie montante du V, vous produirez les documents à destination de la formation, l'utilisation, la maintenance, l'installation et les essais.



Pour rester fidèle aux objectifs initiaux du projet, les étapes situées à droite du V devront être menées en cohérence avec les étapes situées à gauche du V, grâce à l'utilisation des exigences produites en amont. C'est ce que l'on appelle la *traçabilité* en ingénierie système. Le chapitre 3 étudie en détail les exigences et la traçabilité.



## Gérer la complexité grâce aux modèles

Certains modèles sont utiles, surtout lorsque vous travaillez sur des projets complexes. Si vous pouvez mettre au point des méthodes de conception, de test et de vérification de votre système *avant* de démarrer son implémentation, vous vous épargnerez beaucoup d'efforts et vous consoliderez votre rôle dans le projet ! L'une des méthodes possibles consiste à utiliser des modèles pour concevoir et détailler votre système au cours du processus de développement.



La modélisation de système vous permet de capturer la complexité à différents niveaux, dont les systèmes de systèmes (aussi appelés « écosystèmes »), systèmes, sous-systèmes et composants. Elles vous permettent d'étudier en détail chacun de ces niveaux (*les niveaux d'abstraction*) de manière indépendante. Vous pourrez également masquer ou afficher les détails selon vos besoins..

La modélisation peut se présenter sous de nombreuses formes. Le niveau le plus simple peut être un tableur utilisé pour calculer certaines propriétés empiriques. Mais cela peut aller jusqu'à une simulation informatique interactive très complexe.

Imaginons, par exemple, que vous cherchiez à comprendre comment le système automobile (une voiture en somme) que vous développez traite les données de collision et les transmet au service d'intervention. Vous devez étudier les données relatives aux capteurs de la voiture et la manière dont celle-ci interagit avec le système externe. Vous ne voulez cependant pas être gêné par des détails superflus tels que les schémas de câblage ou la position des composants. Le bon modèle ne vous montrera que ce dont vous avez besoin.



Les modèles vous apportent les avantages suivants :

- Ils permettent de vous concentrer sur les données importantes dans votre travail en cours tout en assurant une parfaite homogénéité au niveau de la conception.
- Ils capturent la structure (architecture) et le comportement (fonctionnalités) d'un système, illustrant ainsi les relations et les interactions entre les composants du système.

- ✔ Vous pouvez utiliser les modèles pour comprendre les comportements à différents niveaux d'abstraction, et explorer des alternatives en terme d'architecture dès les premières étapes du processus de développement, ce qui vous permettra de choisir l'architecture optimale.
- ✔ Vous pouvez mettre au point des modèles exécutables qui vous permettront de valider votre conception, en les confrontant aux exigences initiales.
- ✔ Les modèles vous permettent de montrer les détails d'un système selon différents points de vue (par ex. : architecture, logique, fonctionnalité, physique, données et utilisateur). Vous pouvez ainsi répondre à des problématiques variées.
- ✔ Vous pouvez utiliser les modèles pour mieux gérer vos efforts en termes de développement.
- ✔ La modélisation offre une grande visibilité dans la conception d'un système. Vous pouvez ainsi mener des discussions très ciblées, favorisant la compréhension mutuelle.
- ✔ Les modèles permettent de partager réflexions et critères de décision, ce qui facilite la collaboration à l'intérieur de l'équipe.
- ✔ Plus important encore, la modélisation de système offre un point de synchronisation entre les différentes disciplines d'ingénierie pour répondre en particulier à une question essentielle dans le développement des systèmes intelligents : comment coordonner matériel et logiciel ?

## Parler la même langue

De même qu'un architecte utilise un ensemble prédéfini de symboles pour représenter les éléments de construction dans un dessin ce qui permettra au maître d'ouvrage de l'interpréter correctement, l'équipe de développement doit utiliser un langage commun pour représenter les conceptions afin de permettre une compréhension partagée.

En 2001, l'International Council on Systems Engineering (INCOSE) ainsi que l'Object Management Group (OMG) ont mis en place un programme destiné à promouvoir un langage commun pour la modélisation de systèmes. Quelques années plus tard, le langage SysML (System Modeling Language) était né. Développé sur la base du langage UML (Unified Modeling Language) pour le développement du logiciel, SysML est un langage destiné à la modélisation des systèmes et des systèmes de systèmes.

SysML repose sur un ensemble réduit de diagrammes. Il est d'ailleurs considérés par certains comme tellement simple qu'ils l'assimilent à des peintures rupestres ! Il se base sur la notion de bloc qui peut être utilisée pour représenter du matériel, du logiciel, des installations, du personnel ou tout autre élément de système. Vous définissez la structure interne et l'utilisation prévue d'un élément donné (par ex. : le freinage ABS) en construisant une hiérarchie de *diagrammes structurels*. Ensuite, grâce à une autre hiérarchie de *diagrammes comportementaux*, vous montrez l'interaction des éléments entre eux et avec les *acteurs* (utilisateurs, systèmes externes ou environnement). Ces échanges sont nécessaires à l'accomplissement ou à la *réalisation* du comportement visé.

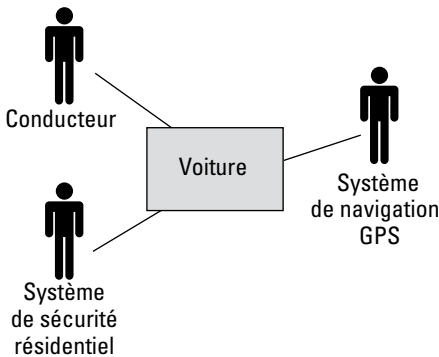
Outre la modélisation de la structure (architecture) et le comportement dynamique (fonctionnalités) du système, SysML permet de modéliser les exigences et les paramètres de performance. Ainsi, dans le cas d'un système automobile par exemple, vous pouvez créer un diagramme d'exigence pour définir une contrainte, par exemple, « atteindre l'arrêt complet en partant d'une vitesse de 105 km/h sur une distance inférieure à 55 m sur une surface propre et sèche ». Vous pourrez également réaliser les diagrammes paramétriques détaillant les équations régissant le mouvement de la voiture. De plus, vous pouvez utiliser de nouveaux outils logiciels de simulation très puissants, qui vous font croire que vous conduisez une nouvelle voiture de course. Cela équivaldrait à aller faire un essai sur route pour tester la tenue de la voiture avant même qu'elle ne soit construite.



En proposant un ensemble complet d'éléments de modélisation, SysML force les ingénieurs systèmes et les architectes à être clairs et précis dans leurs travaux. Cela réduit les ambiguïtés et favorise la qualité tout en raccourcissant les cycles de développement et les coûts.

## ***Une image vaut mille (voire un million de) mots***

Pour mieux comprendre l'utilisation de SysML, il est sans doute préférable de jeter un œil à un de ces diagrammes. La Figure 2-3 présente un *diagramme contextuel* simple montrant une voiture moderne dotée d'un système de navigation GPS et d'une télécommande pilotant le système de sécurité résidentiel.



**Figure 2-3 :** diagramme contextuel pour une voiture.

Le diagramme contextuel vous permet de définir les frontières, ou le contexte, de votre système. Ici, le contexte est la voiture. Il interagit avec trois *acteurs* extérieurs au système : le conducteur, le système de navigation GPS et le système de sécurité résidentiel. Le terme « acteur » désigne toute personne interagissant avec le système, qu'il s'agisse d'un utilisateur, d'un autre système ou de l'environnement. Le diagramme contextuel permet de définir le champ d'application du système en question.



En définissant le système et ses acteurs, vous identifiez les interactions significatives, et donc les exigences et les interfaces nécessaires. Vous pouvez ensuite définir les caractéristiques des interfaces et les flux de données entre le système et ses acteurs.



Sans diagramme contextuel, vous risquez d'oublier l'un ou l'autre acteur ou passez à côté d'exigences pour le système. Ainsi, si vous ne définissez pas le système de sécurité résidentiel parmi les acteurs de l'écosystème de votre voiture, celle-ci ne sera pas suffisamment intelligente pour armer votre système d'alarme.

## Chapitre 3

# Révolution des exigences

### *Dans ce chapitre*

- ▶ Voir le côté positif du changement
- ▶ Couvrir toutes les exigences en incluant les cas d'utilisation
- ▶ Répercuter les exigences sur tout le processus de développement
- ▶ Analyser les impacts du changement
- ▶ Maîtriser la gestion des exigences

Il n'y a pas si longtemps, les exigences étaient toutes établies en début de projet. Il fallait ensuite concevoir le produit afin qu'il réponde à ces exigences et lorsque les commerciaux revenaient vers vous pour vous signaler que le client voulait une modification (voire deux ou trois), vous hésitez un peu avant de leur répondre qu'il s'agissait d'un changement coûteux qui nécessiterait au moins 57 signatures.

Cette époque est bel et bien révolue. Dans un marché volatil et concurrentiel où la réussite dépend de votre capacité à satisfaire pleinement les clients ; deux choix s'offrent à vous : changer ou subir les événements. Dans ce chapitre, vous découvrirez comment concevoir les exigences en gardant les modifications futures à l'esprit et comment faire en sorte que votre conception y réponde.

## Embrasser la philosophie du changement

À l'époque des produits intelligents, vous devez pouvoir réagir rapidement à la dynamique du marché, à l'évolution des besoins des clients, aux nouveaux concurrents et aux dernières réglementations. Les concepteurs de produit ont constaté au fil des ans que les exigences *devaient* évoluer au fur et à mesure de la compréhension des besoins au cours du processus de développement. Cependant, le changement est l'ennemi du processus de développement traditionnel. Que faire donc ?

Trouver un nouveau processus.

Vous serez sans doute heureux d'apprendre que d'autres sont passés par là avant vous et ont mis au point un tout nouveau processus d'*ingénierie des exigences*. Bien au-delà d'une analyse préliminaire avec une définition trop simpliste des exigences, l'ingénierie des exigences se veut une approche complète couvrant à la fois la définition des besoins et des tests, afin de faciliter le changement.



Les exigences sont d'autant meilleures lorsqu'elles sont pensées pour évoluer. La philosophie de l'ingénierie des exigences vise à accueillir favorablement le changement. Le changement devient un objectif en soi !

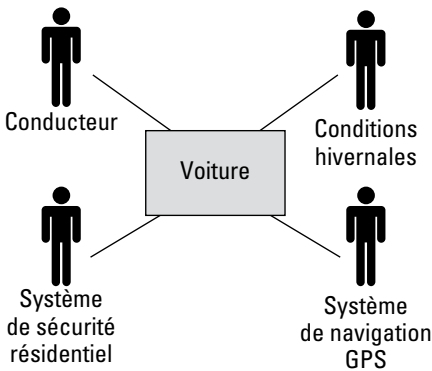
## Comprendre le contexte

Avant de définir un ensemble d'exigences spécifiques à un produit intelligent ou à un système, vous avez tout intérêt à réfléchir au problème que votre produit tente de résoudre. Ainsi, si vous comptez fabriquer une voiture, il est important que vous sachiez exactement comment elle sera utilisée et par qui. Allez-vous privilégier la conduite en ville, les longs trajets ou des circuits sportifs ? Le marché cible se compose-t-il essentiellement de conducteurs âgés, de jeunes ou de parents au foyer ? La voiture sera-t-elle soumise à des conditions difficiles (froid extrême, routes sablées, chaleur extrême, routes montagneuses, altitude) ?



La compréhension du *contexte* est essentielle pour mettre au point des systèmes permettant d'atteindre les objectifs de marketing et commerciaux. Le terme « contexte » désigne un ensemble d'*acteurs* (par ex. : les utilisateurs, les autres systèmes et l'environnement) avec lesquels le système interagit et comment ces interactions entre le système et les acteurs ont lieu.

La Figure 3-1 présente un diagramme contextuel pour une voiture dotée d'un système de navigation GPS et d'un système de sécurité résidentiel qui sera vendue dans les pays froids. Dans ce schéma, la voiture est en fait une boîte noire qui interagit avec les acteurs suivants : le conducteur, le système de navigation GPS, le système de sécurité résidentiel et les conditions hivernales. La définition du contexte permet de comprendre quelles fonctions sont nécessaires et quelle est la nature des échanges entre le système et ses acteurs.

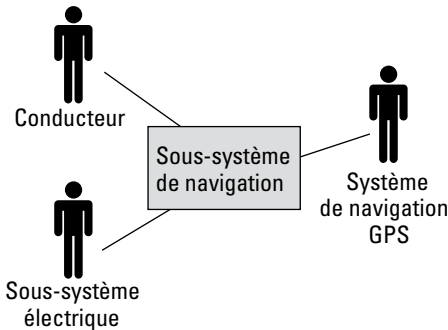


**Figure 3-1 :** le contexte d'un système en définit les limites et les interfaces.

La compréhension du contexte permet d'envisager toutes les exigences, relations et interfaces nécessaires avant de démarrer le processus de développement. Ainsi, si vous omettez le facteur « Conditions hivernales », vous risquez de ne pas proposer de « pack hiver » composé de câbles de batterie et d'un revêtement spécifique pour le châssis.

Au niveau d'abstraction le plus haut, votre système est une boîte noire qui interagit avec un ensemble d'acteurs externes. Si vous jetez un œil à l'intérieur de cette boîte noire, vous y verrez un ensemble de sous-systèmes interconnectés (par ex. : système de freinage antiblocage, système de navigation, etc.) qui, ensemble, forment votre système. Vous pouvez ensuite décomposer chaque sous-système en composants interconnectés, voire en sous-sous-systèmes. À chaque *niveau de décomposition* (par ex. : système, sous-système, composant) de votre système, le contexte change.

Comparez le diagramme contextuel de la voiture (v. Figure 3-1) avec le diagramme contextuel du sous-système de navigation intégré (v. Figure 3-2). Au niveau du système, la voiture est la boîte noire qui interagit avec les acteurs externes. Si l'on passe au niveau de décomposition suivant, le sous-système de navigation devient la boîte noire. Il interagit avec un autre groupe d'acteurs, à savoir le conducteur, le sous-système électrique de la voiture et le système GPS.



**Figure 3-2 :** le contexte change au fur et à mesure que vous explorez les différents niveaux du système.



La compréhension du contexte permet d'établir les limites du système et d'en définir les interfaces. Vous disposez ainsi de tous les atouts pour une spécification précise des exigences du système.

## Se plonger dans les exigences

Pensez les exigences selon trois grandes catégories ou couches :

- ✓ **Exigences sources.** Il s'agit des exigences émanant des clients ou des parties prenantes. Elles peuvent être générales, détaillées, spécifiques, complètes ou partielles, voire parfois un peu de tout cela. Comme le dit le proverbe, « le client est roi ». Il en va donc de même pour ses exigences. Vous devrez faire avec ce qu'il vous donne.
- ✓ **Exigences liées à la mission ou à son objectif marketing.** Il s'agit des exigences qui définissent le contexte opérationnel dans lequel le système évolue, non pas ce qu'il fait, mais le rôle qu'il joue dans son environnement. Si l'on prend l'exemple d'un nouvel avion militaire, il s'agira notamment d'une description des missions qu'il assurera. Dans le cas d'un nouveau smartphone, les exigences marketing décriront son fonctionnement au sein de l'infrastructure de communication de l'opérateur.



✓ **Exigences du système/sous-système.** Ces exigences établissent ce que le système doit pouvoir faire. Elles commencent aux niveaux supérieurs et sont ensuite décomposées pour définir les exigences des sous-systèmes inférieurs. Elles peuvent être exprimées sous forme d'énoncés décrivant les différentes fonctionnalités ou sous forme de modélisations et diagrammes.

Pour chaque niveau d'abstraction, les exigences définissent ce qu'un système doit pouvoir faire et comment il doit le faire. Elles n'établissent cependant pas comment il doit être développé. Étant donné que les systèmes intelligents se font de plus en plus complexes, il est quasi impossible d'obtenir la liste complète des exigences dès le début du projet. Et dans un environnement où le changement empêche la stabilité, comme dans le secteur de l'électronique, mieux vaut ne pas figer les exigences trop tôt.

## Obtenir un avis plus large

Il est bon de récupérer un maximum d'informations pour définir les exigences sources. Pour cela, faites appel à un maximum de parties prenantes le plus tôt possible lors du processus d'ingénierie des exigences. Vous commencerez bien sûr par vos clients, mais vous chercherez aussi à obtenir des informations sur les lois et normes de sécurité ou industrielles régissant le système, les interfaces et les échanges de données avec les autres systèmes (par ex. : le système GPS) ainsi que les contraintes commerciales et de marketing.



Vous privilégiez les exigences décrivant les capacités du système plutôt que ses fonctions. (Par exemple, ce que le système doit faire et comment il doit le faire.) Si un client décrit une fonction, demandez-lui pourquoi il souhaite cela. Cela devrait permettre d'en extraire la capacité.

Il est important aussi de définir les exigences *fonctionnelles* et *non fonctionnelles*. Les exigences fonctionnelles décrivent ce que le système doit faire sur la base de certains éléments. Par exemple, avec un point de départ et un point de destination, votre système de navigation devra définir un itinéraire et l'afficher. Les exigences non fonctionnelles vous permettent de définir les critères de performance et de qualité, voire d'imposer certaines contraintes lors de la conception. Cela comprend notamment les exigences en termes de vitesse, capacité, fiabilité, poids, utilisation et évolutivité.

Pour être sûr de couvrir toutes les bases, vous préparerez des cas *d'utilisation* décrivant tous les scénarios d'utilisation possible des fonctions du système. Par exemple, la fonction « définir un itinéraire » de votre système de navigation permet de « trouver la station-service la plus proche » ou « afficher les hôtels à proximité de la destination ». Les cas d'utilisation sont généralement des séquences composées d'au moins une fonction. Les cas d'utilisation racontent des histoires concrètes sur l'utilisation du système. Ils peuvent être utilisés pour les trois types d'exigence (source, mission/marketing et système/sous-système).



En définissant les usages au fur et à mesure que vous fixez les exigences, vous parviendrez à concevoir un système apportant une véritable valeur ajoutée.

## Satisfaire et dériver les exigences

Après avoir défini le contexte de votre système et un ensemble d'exigences de haut niveau, utilisez-les pour établir la conception de haut niveau (v. Figure 3-3). Au cours de cette étape de conception, vous définissez des exigences complémentaires, comme le plan de vérification du système qui devra être respecté pendant les tests, ou les contraintes structurelles qui devront être respectées en conception détaillée.

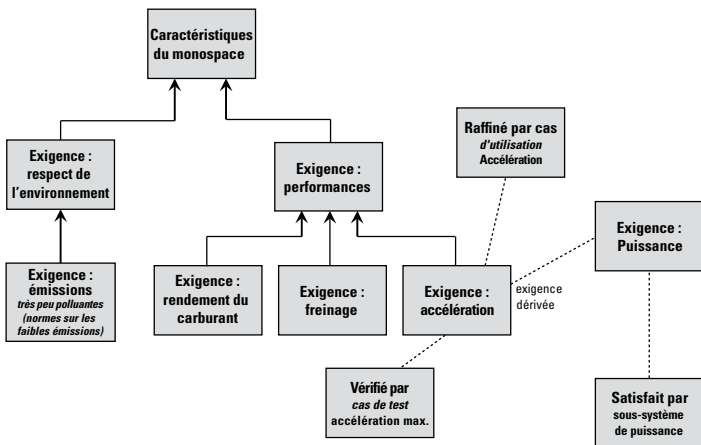


Figure 3-3 : le processus de conception de haut niveau.



Lorsque vous avancez dans la conception détaillée de votre système, vous devez répondre aux exigences définies au cours des étapes précédentes (niveaux supérieurs). Vous en dérivez aussi les nouvelles exigences nécessaires aux niveaux actuels et suivants (inférieurs).

Comme vous pouvez le constater, la définition du terme « exigences » s'élargit. La séparation entre exigences et conception devient de moins en moins marquée tandis que les liens entre test et conception prennent de plus en plus d'importance. La liste définitive de vos exigences est une combinaison des exigences initiales pour le système de niveau supérieur et de celles acquises pendant le processus de conception.

## *Créer des diagrammes contenant les exigences*

Bien que les exigences écrites selon les méthodes traditionnelles basées sur le texte restent nécessaires pour les gros projets contraints par des obligations contractuelles, elles ne suffisent généralement pas pour concevoir des produits intelligents complexes, polyvalents et fortement interactifs. Heureusement, le langage de modélisation système (SysML) a été largement accepté, ce qui permet de disposer d'un cadre pour l'ingénierie des exigences. Même si un grand nombre d'exigences existent peut-être déjà dans une base de données, en particulier celles provenant du client, il y a de nouveaux moyens plus efficaces pour visualiser tout cela et travailler sur ces ensembles d'exigences.

Le langage SysML vous permet de créer des diagrammes hiérarchiques d'exigences montrant les dépendances, leur classification en originales ou dérivées et les justifications des choix de conception. Dans notre exemple, il y a deux exigences de haut niveau : Respect de l'environnement et Performances. (v. Figure 3-3.) L'exigence Respect de l'environnement contient une sous-exigence intitulée Emissions.

L'exigence Performances contient trois sous-exigences, l'accélération est l'une d'entre elles. L'exigence Accélération est ensuite raffinée par des cas d'utilisation spécifiques. Un programme de test de l'accélération maximale est défini (le plan de test devra vérifier cette exigence). Enfin, elle produit une exigence dérivée relative à la puissance, qui doit être satisfaite par le sous-système de puissance.

Les diagrammes SysML vous permettent de voir plus facilement les relations entre exigences et de répondre à des questions telles que « À quelles exigences le sous-système de motorisation répond-il ? » ou « Quel serait l'impact d'une modification du volume de chargement ? ». Lorsque vous traitez les exigences comme faisant partie intégrante de l'architecture du système, il est bien plus facile de visualiser l'impact des changements sur celui-ci. Et n'est-ce pas là l'un des objectifs de l'ingénierie système ?

## *Baser les décisions de conception sur les exigences*

Les exigences se répercutant sur tout le processus de conception, la nature de certaines d'entre elles peut donner lieu à certains choix spécifiques. Par exemple, si vous travaillez sur un système d'évitement de collision d'avion, dès le début du projet, vous disposerez de trois options qui répondent chacune à l'exigence fonctionnelle, à savoir éviter les collisions :

- ✔ Système radar embarqué sophistiqué
- ✔ Système de transpondeur
- ✔ Système essentiellement manuel (communication avec les contrôleurs)

Chaque option implique un ensemble de composants précis, des contraintes d'espace et des coûts. Elles comportent également un certain niveau de collaboration entre le système et ses acteurs (par ex. : les contrôleurs du trafic aérien). Tandis que vous décomposez votre architecture, vous pourriez être confronté à une exigence non fonctionnelle, par exemple relative à l'espace maximal disponible dans le cockpit, susceptible de limiter vos options en termes de conception. Le cas échéant, le radar risque de ne pas pouvoir être installé !

Les architectes des systèmes utilisent une technique appelée *étude comparative* pour évaluer les différentes possibilités et faire les bons choix. C'est un peu ce que nous faisons tous lorsque nous devons prendre une décision importante. Il s'agit donc d'identifier les critères importants de décision, d'évaluer chaque alternative pour chacun des critères, puis de calculer la moyenne pondérée, ce qui permet de prendre une décision justifiée. Les études comparatives exigent souvent de longues recherches pour passer en revue la totalité des alternatives.

Le suivi des exigences et de leur influence sur les autres parties du système constitue un mal nécessaire si vous voulez fournir des produits efficaces. Des progiciels peuvent vous aider dans cette tâche.



Si vous oubliez une exigence, vous risquez de rencontrer de gros problèmes. Vous pouvez ainsi concevoir le porte-gobelet pour voiture le plus original au monde, si vous le placez sous la radio et que votre grand café crème bloque l'accès aux commandes parce que vous avez omis l'exigence « laisser suffisamment de place » lors de la conception, vous aurez raté votre produit. De telles erreurs peuvent réellement vous coûter cher.

## *Jongler avec les exigences et les conceptions*

Dans l'exemple du système d'évitement de collision d'avion évoqué plus haut, imaginons que vous sélectionniez l'option « système de transpondeur ». Cette option semi-automatique implique le contrôle des signaux du transpondeur des appareils évoluant dans la zone. Ce choix va vous amener à produire des sous-exigences au niveau sous-système Transpondeur, ainsi que son plan de test.

Vous êtes déjà bien avancé dans la conception de ce sous-système lorsque vous recevez une demande de modification des exigences. L'appareil doit désormais pouvoir éviter les collisions de manière autonome. Votre choix ne répond pas à ces critères. Vous devez donc l'abandonner et envisager une nouvelle solution : le radar.

Certains changements peuvent donner lieu à des exigences dérivées sur la conception d'un sous-système, qui peuvent aller jusqu'à influencer la conception même du système. Par exemple, les contraintes de coût pourront limiter le choix d'un composant du système, ce qui pourrait restreindre les fonctionnalités du sous-système et donc influencer les exigences originelles. Il y a quelques années, ce processus de propagation des changements prenait des heures, voire des semaines. Les ingénieurs devaient assurer le suivi et modifier les documents de spécification et de conception rédigés au moyen d'outils de traitement de texte et de tableurs. L'introduction de la modélisation dans le processus apporte ici beaucoup de facilité, comme nous le verrons au chapitre 4.



Mettez en place un processus formel de modification des exigences afin de mieux maîtriser l'impact des modifications sur la conception de votre système.

## Associer les exigences aux tests

Vous venez de concevoir un nouveau système GPS ultra-réactif pour les voitures. Le prototype a été conçu et votre équipe de test vous signale que le système fonctionne à merveille. En fait, il calcule l'itinéraire entre deux points à une telle vitesse (grâce à votre brillant algorithme) qu'il surpassera forcément ses concurrents. Votre patron est tellement fier qu'il propose le prototype à votre PDG pour qu'elle puisse l'utiliser pendant la journée. La PDG est impressionnée, du moins jusqu'à ce qu'elle tente de trouver le restaurant chinois le plus proche. Elle perd patience tant l'attente est longue.

Vous vous demandez ce qu'il s'est passé. Vous réalisez alors que votre algorithme convient parfaitement aux itinéraires reliant un point A à un point B, mais pas pour trouver une destination dans un rayon donné. Bien que l'équipe de test ait passé cette fonction au crible, elle n'a pas pu tester tous les cas d'utilisation (peut-être a-t-on même oublié de spécifier ce cas d'utilisation).



La création de liens entre exigences et tests constitue une activité clé dans le processus d'ingénierie système. Vous devez vous assurer que le système conçu correspond à celui prévu, et pas seulement qu'il constitue un ensemble « qui fonctionne ».

À chaque niveau de décomposition, tandis que vous affinez et dérivez les exigences, vous devez aussi créer les jeux de tests destinés à vérifier si le système répond aux attentes. Si les exigences varient, vous devez modifier les jeux de test. Souvent, le fait de spécifier les critères de test pour un ensemble d'exigences vous aide à améliorer et affiner les exigences elles-mêmes.



Une bonne pratique est de rédiger les exigences de manière à s'assurer qu'elles *peuvent* être testées. Veillez aussi à établir le plus tôt possible *comment* elles le seront.

## Veillez à laisser une trace

Pour que toutes les exigences soient correctement implémentées, vous devez pouvoir suivre chacune d'elle tout au long du processus de développement et de test.



La traçabilité des exigences représente la capacité à associer chacune d'elle à trois éléments connexes :

- ✓ Les besoins des parties prenantes (exigences sources) qu'elle satisfait
- ✓ Les éléments du système qui la réalisent
- ✓ Les tests qui permettent de la vérifier

La traçabilité vous aide à respecter les règlements et normes, à éviter de passer à côté d'exigences et de rester concentré sur les objectifs principaux du projet. En établissant des liens de traçabilité de bout en bout, vous pourrez identifier avec précision les exigences impactées par les dernières modifications ainsi que les choix d'architecture associés, avant même de faire les changements en question.

## Un dur labeur récompensé

Pour des grands systèmes complexes, la gestion des exigences peut s'avérer être un cauchemar. La plupart des équipes de développement se composent de centaines, voire de milliers d'architectes et ingénieurs qui interviennent tous sur les exigences, qu'il s'agisse de les créer, de les modifier ou simplement de les passer en revue pour mieux les comprendre. La traçabilité couvre généralement quatre niveaux de décomposition, partant des exigences des parties prenantes, puis traversant le niveau de conception Système et le niveau de conception Composants, pour arriver jusqu'aux tests. Cette traçabilité va même au-delà de ces niveaux de décomposition pour aller jusqu'à la chaîne logistique, ce qui complique davantage les choses.

Une ingénierie efficace des exigences implique de nombreuses disciplines : conception des systèmes, architecture, logiciel, mécanique, électricité et tests. Les fonctions commerciales telles que le marketing et l'approvisionnement peuvent également être partie prenante dans la gestion des exigences.



Les outils logiciels vous aident à mieux supporter le processus de gestion des exigences. De tel outils sont conçus pour conserver les historiques (par exemple de choix de conception), tracer les changements, analyser leur impact et automatiser le processus de gestion des changements. Ils vous signalent également si vous avez oublié certaines exigences, réalisé un modèle trop complexe ou introduit des non-conformités.

La gestion des exigences est une activité particulièrement difficile, mais l'usage d'outils adéquats peut s'avérer très rentable en termes de coûts et de planning, vous assurant ainsi la réussite du projet et moins de stress.



## Chapitre 4

# Abstraire le système par la modélisation

### *Dans ce chapitre*

- ▶ Décomposer le système en niveaux d'abstraction
- ▶ Visualiser l'assemblage des composants du système
- ▶ Se représenter le comportement du système
- ▶ Raffiner les modèles du système par itération

Une formule est une sorte de modèle. Elle englobe les rapports mathématiques entre les variables d'entrée et les représente sous la forme d'une structure mathématique qui s'applique à une multitude de données d'entrées. La visualisation de la formule permet de mieux comprendre les rapports entre les différents éléments de la structure. Enfin l'utilisation d'une formule permet de tester différentes possibilités jusqu'à ce que vous trouviez celle qui vous convient.

Dans ce chapitre, vous apprendrez à utiliser la modélisation de système pour gérer sa complexité et montrer les relations principales entre composants de système. Vous verrez aussi comment vérifier un système à moindre frais avant même son implémentation.

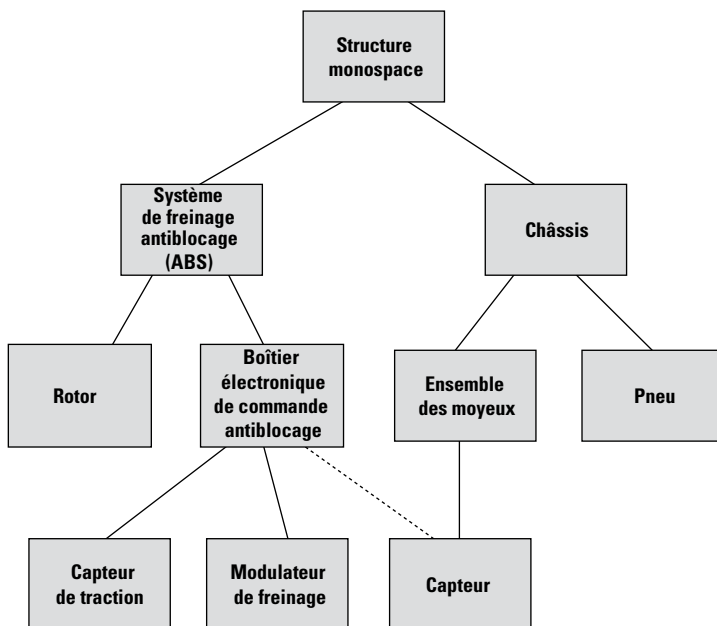
## *Modélisation de l'architecture d'un système*

Les modèles architecturaux vous permettent de capturer la nature statique de votre système, à savoir sa structure et l'utilisation envisagée. Prenez par exemple, l'architecture simplifiée d'une voiture présentée aux Figures 4-1 et 4-2.

Le modèle structurel partiel de la Figure 4-1 décompose le système en plusieurs niveaux. Le niveau supérieur représente le système global, à savoir la voiture. En descendant d'un niveau, nous retrouvons deux sous-systèmes parmi les nombreux qui composent une voiture : le système de freinage antiblochage (ABS) et le châssis.

Le sous-système du châssis se décompose en un sous-système inférieur (l'ensemble des moyeux), un composant mécanique (les pneus) et d'autres sous-systèmes et composants qui ne sont pas présentés dans ce schéma. Le sous-système Ensemble des moyeux se compose d'un élément électronique (capteur) et de plusieurs éléments mécaniques qui ne figurent pas sur l'illustration.

Le sous-système ABS est formé d'un composant mécanique (rotor) et d'un autre sous-système (le boîtier électronique de commande antiblochage). Le sous-système du boîtier électronique de commande antiblochage est quant à lui composé de deux éléments électroniques : le capteur de traction et le modulateur de freinage. La ligne pointillée reliant le capteur de l'ensemble des moyeux au boîtier électronique de commande antiblochage montre la relation qui existe entre les deux, même si le capteur n'est pas un composant du sous-système du boîtier électronique de commande antiblochage.

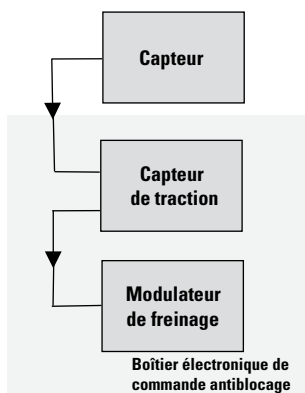


**Figure 4-1 :** modèle d'architecture simplifiée pour un monospace.



Les modèles structurels, comme l'architecture présentée à la Figure 4-1, présentent la hiérarchie du système et les relations entre les composants du modèle.

La Figure 4-2 présente un autre élément de l'architecture, à savoir l'interaction entre les composants du sous-système du boîtier électronique de commande antiblocage et leur lien avec le capteur de l'ensemble des moyeux. Dans ce cas, la sortie du capteur est reliée à l'entrée du capteur de traction, dont la sortie est reliée à l'entrée du modulateur de freinage. Ce diagramme simple présente les connexions indiquant l'usage *prévu*, mais il ne précise pas les conditions dans lesquelles ces interactions ont lieu. Vous devrez utiliser un modèle de comportement (que nous évoquerons dans la rubrique suivante) pour décrire la séquence d'événements à l'origine des interactions, par exemple, le capteur indique une perte de traction, ce qui provoque le déclenchement du modulateur de freinage par le capteur de traction.



**Figure 4-2 :** les modèles d'architecture montrent les flux d'information et de contrôle.

Les modèles structurels présentent l'architecture physique, comme le montrent les exemples ci-dessus. Ils peuvent aussi être utilisés pour présenter l'architecture logique. Les architectures logiques sont de plus en plus souvent utilisées pour réaliser des systèmes complexes étant donné qu'ils permettent aux ingénieurs de raisonner sur les interactions fonctionnelles sans définir une structure physique donnée. Prenons le cas de la voiture de la Figure 4-1. Il y a quelques dizaines d'années, les sous-systèmes qui la composaient, à savoir les freins, le moteur et la radio, étaient totalement séparés. L'architecture physique suffisait pour les comprendre et les fabriquer. Aujourd'hui, l'architecture logique d'une voiture contient des éléments tels que :

- La propulsion, qui interagit aujourd'hui avec le freinage, l'électricité, le moteur ou les régulations.
- Le système de gestion de l'information, qui comprend toutes les données reçues, stockées, envoyées et utilisées dans la voiture.
- Le système de divertissement, dont certains composants peuvent être communs aux systèmes de propulsion et d'information.
- La sécurité, qui dépend de nombreux autres systèmes du véhicule dont le freinage, la gestion de l'information et l'interface utilisateur du conducteur.

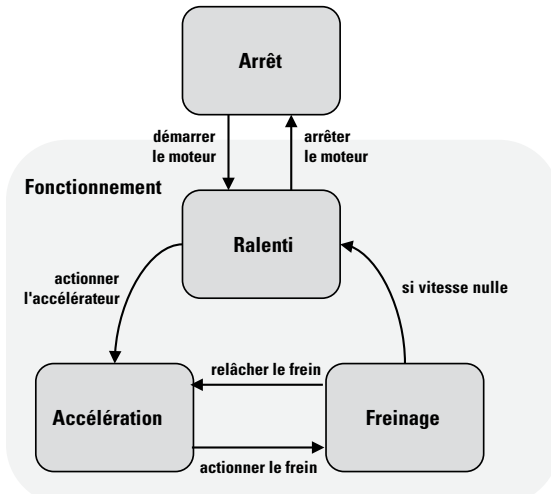
En traitant l'architecture logique séparément - idéalement avant l'implémentation physique -, les ingénieurs Systèmes opteront pour des approches plus efficaces et plus élégantes. Cette double approche couvrant les aspects logiques et physiques permet de mieux gérer la complexité des nouveaux produits intelligents intégrés qui, vous l'avez compris, n'ont plus rien à voir avec l'antiquité que conduisait votre grand-père !

## *Modélisation du comportement du système*

Pour mieux comprendre le comportement du système, vous devez comprendre comment les composants de l'architecture (logique et physique) d'un système interagissent. Les modèles de comportement capturent les informations dynamiques d'un système, incluant les changements d'état, les actions accomplies suite à la réception de certains événements spécifiques et les interactions entre les différents éléments. Les modélisations intègrent également les flux de données et de contrôle entre les différentes activités, notamment la manière dont les données de sortie des capteurs sont transmises aux opérations du boîtier électronique de commande antiblocage ou la manière dont la commande est transmise d'un composant à un autre.

La Figure 4-3 présente un diagramme d'état simplifié décrivant les états de fonctionnement d'une voiture. La voiture adopte l'un de ces quatre états (arrêt, ralenti, accélération, freinage) jusqu'à l'apparition d'un événement spécifique (par ex. : « Actionner le frein ») qui amène la voiture à passer d'un état à un autre.

Vous pouvez établir des liens entre les modèles comportementaux et structurels afin d'assurer une certaine homogénéité grâce à un procédé appelé « *allocation* ». Par exemple, vous allouez l'action « Détecter la perte de traction » au composant « Capteur de traction » de votre modélisation structurelle. L'action « Moduler la force de freinage » en revanche sera allouée au composant « Modulateur de freinage ».



**Figure 4-3 :** modèle comportemental montrant les transitions entre les différents états de fonctionnement.

Pour les systèmes complexes, vous aurez besoin de plusieurs petits modèles comportementaux pour représenter toute la dynamique. La plupart du temps, une activité déclenche une action dans une autre partie du système. Vos modèles individuels sont donc interconnectés. Au terme de la modélisation des activités, vous disposez d'un grand modèle complexe, uniforme et homogène représentant le comportement du système dans sa globalité.



Grâce à des constructions et à une sémantique propres, comme celle fournie par le langage SysML, vous pouvez utiliser des outils logiciels commerciaux pour exécuter les modèles comportementaux. Ces outils sont capables de transformer automatiquement les constructions modélisées en code « if-then-execute ». C'est notamment le cas pour les diagrammes de transition d'état. Ainsi, vous pouvez simuler le comportement du système par logiciel, ce qui vous permet d'étudier des hypothèses, explorer des alternatives et réaliser des analyses d'impact avant même d'implémenter votre système. Les études comparatives qui prenaient auparavant plusieurs heures, voire plusieurs jours, peuvent désormais être réalisées en quelques minutes seulement.

## Construire les modèles

Les standard de l'industrie, comme le langage SysML, permettent de mieux comprendre la modélisation des systèmes tout au long de leur ingénierie. Les

modèles SysML se composent d'un ensemble de *diagrammes* représentant la structure, le comportement, les exigences et les contraintes quantitatives d'un système. Les diagrammes SysML se présentent de quatre manières différentes :

- ✓ **Structure** : ils décrivent les éléments structurels (logiques et physiques), appelés « *blocs* », contenus dans le système ou le sous-système ainsi que les liens entre eux. Par exemple, le sous-système de freinage antiblocage contient un capteur de traction et un modulateur de freinage.
- ✓ **Comportement** : ils décrivent le comportement du système ou d'un sous-système, avec ses changements d'état, les séquences d'activités, les fonctions et les interactions. Par exemple, l'état « Détecter perte de traction » active la fonction « Moduler la force de freinage ».
- ✓ **Exigences** : ils décrivent les exigences précises associées au système ou aux sous-systèmes. Par exemple, un diagramme contiendra le détail des distances de freinage.
- ✓ **Paramétrique** : ils décrivent les contraintes paramétriques du système ou sous-systèmes. Par exemple, l'équation de la force de freinage peut être représentée par une équation paramétrique.

## Exploration des quatre étapes de la modélisation de système

Par nature, les systèmes sont des structures récursives composées de plusieurs niveaux, à savoir un système global à la racine qui se décompose en sous-systèmes, puis en composants. Le meilleur moyen de modéliser un système complet consiste à utiliser un processus récursif (en commençant par la racine) composé de quatre étapes. Les étapes sont représentées par l'anagramme CURE – pensez-y comme un remède à la complexité :

- ✓ **Contexte** : cette étape consiste à définir les limites de votre système en identifiant les personnes et systèmes avec lesquels il interagit (les acteurs). Elle décrit aussi les interfaces, à savoir les modes de communication et les données échangées. Pris dans leur ensemble, les éléments de ce modèle contextuel sont appelés *entreprise*, à savoir le système et son environnement immédiat. Utilisez les diagrammes SysML pour cela.
- ✓ **Utilisation** : cette étape décrit les différents moyens d'utilisation du système par les acteurs en indiquant qui utilisent le système et ce que celui-ci utilise. Elle est la plus efficace lorsqu'elle est décrite comme

une histoire racontée pas-à-pas, en utilisant les cas d'utilisation et détaillée à l'aide de diagrammes d'activité SysML. Raffinez vos exigences et faites en sorte qu'elles soient suffisamment précises et exhaustives (rappelez-vous : beaucoup d'aspects importants font défaut dans les exigences de départ). Ces scénarios d'utilisation vont plus tard former la base de test puisqu'ils s'appuient sur ce que vous avez appris lors de l'analyse de l'utilisation.

✓ **Réalisation** : définissez les modèles structurels (architecture) et comportementaux (fonctions) qui décrivent ensemble comment l'on arrive à telle ou telle utilisation grâce à une collaboration des composants de l'architecture de système. Le comportement visé prend forme grâce aux différents éléments du système. Cela se distingue fortement des processus de développement traditionnels consistant à allouer les exigences aux composants physiques et espérer que tout fonctionnera lors de l'utilisation réelle. Dans le processus que nous préconisons, les usages sont d'abord définis ensuite il faut concevoir le système sur la base de ces usages de manière à pouvoir *vérifier* qu'il répond bien aux exigences d'utilisation.

✓ **Exécution** : cette étape consiste à exécuter les modèles de comportement pour montrer que votre modélisation répond bien aux exigences. Les modèles simples et exécutables, même à haut niveau d'abstraction, constituent un moyen efficace et rentable de détecter des problèmes complexes, des erreurs de communication, des exigences mal définies ou non prises en compte ainsi que d'autres éléments susceptibles de provoquer des retards sur le planning le plus tôt possible. La modélisation permet à tout le monde d'être sur la même longueur d'onde avant toute implémentation.

Commencez la modélisation par le niveau hiérarchique le plus élevé, à savoir le système (par ex. : un monospace). Après avoir détaillé le contexte et les cas d'utilisation, vous allez décrire l'architecture et le comportement de ses composants sur la base des exigences. Vous devrez ensuite exécuter le modèle afin de vérifier s'il correspond bien aux usages attendus du système. Après avoir finalisé la modélisation au niveau du système, recommencez cette activité pour le niveau de décomposition suivant, à savoir les sous-systèmes.

Continuez à modéliser à chaque niveau de décomposition en changeant de contexte à chaque fois, jusqu'à ce que vous atteigniez le niveau le plus bas (les composants). Vous pourrez alors définir l'implémentation physique de votre modèle (par ex. : électronique, logiciel ou mécanique).

Pour chaque niveau, progressez horizontalement à travers le cycle en V. Effectuez les vérifications et les validations nécessaires en utilisant le modèle comme référence. Utilisez les modèles exécutable et les autres simulations mathématiques pour faire un maximum de vérification avant l'implémentation.



Votre but ultime consiste à réaliser une modélisation de système complète : une sorte de représentation virtuelle du système réel. Ce n'est pas encore possible à ce stade, mais étant donné que les outils de modélisation s'améliorent et commencent petit à petit à intégrer les différentes disciplines d'ingénierie, nous nous en approchons de plus en plus.

## *Comprendre pourquoi la modélisation est à la mode*

Vous pensez sans doute que la modélisation de l'architecture et du comportement d'un système complexe ne mérite pas tant d'efforts. Vous changerez probablement d'avis lorsque vous vous souviendrez des difficultés rencontrées lors de votre dernier projet, lorsque l'équipe de développement est passée à côté d'un élément crucial, menant ainsi le projet à sa perte.

La modélisation vous permet de tenir compte de tous les détails de votre système en les organisant intelligemment. Vous pourrez ainsi visualiser, comprendre et résoudre les complexités de la structure et du comportement du système. Elle vous permet d'explorer des options différentes d'architecture et de conception, d'effectuer des analyses comparatives et d'évaluer l'impact des changements avant même de démarrer l'implémentation de votre système. Vous limitez ainsi les risques et les coûts de développement associés à votre projet.

La modélisation permet de raisonner sur les différents composants du système à des niveaux variés. Vous pouvez ainsi profiter de plusieurs points de vue sur le système : planification, exigences, architecture, conception, implémentation, déploiement, comportement, données d'entrée, données de sortie, etc. Vos collègues et vous pouvez ainsi raisonner aussi facilement sur des questions générales que sur des détails.

Les langages et techniques de modélisation communes au domaine, dont SysML, permettent de réduire l'ambiguïté et d'éliminer la barrière de la langue qui pourrait exister entre les différentes équipes de développement. Ils servent aussi de référentiel unique pour gérer le développement et la documentation du projet. Une meilleure collaboration et une documentation claire favorisent l'efficacité, réduisent les délais de développement et, par-dessus tout, améliorent la qualité.



## Chapitre 5

# Garantir une excellente qualité

### *Dans ce chapitre*

- ▶ Intégrer la qualité au processus de développement des systèmes
- ▶ Combiner les tests et la conception
- ▶ Utiliser la modélisation pour détecter rapidement les erreurs

**A**uparavant, l'assurance-qualité avait lieu à la fin du cycle de développement, comme si la qualité était une chose que l'on pouvait ajouter au produit avant son expédition. Et lorsque des défauts étaient détectés, des efforts considérables étaient nécessaires pour en identifier les causes et réparer les erreurs.

Dans un monde de produits intelligents et complexes de plus en plus nombreux et pilotés par des logiciels, la qualité doit faire partie intégrante du processus de développement afin que vous puissiez garder l'espoir de livrer un produit non défectueux et adéquat à un usage particulier. Dans ce chapitre, vous verrez comment l'ingénierie système permet d'identifier les problèmes de mauvaise qualité grâce à une validation et une vérification au plus tôt, permettant ainsi d'augmenter les chances de réussite.

## *Explorer les niveaux de test*

Lorsque vous concevez un produit intelligent incluant plusieurs sous-systèmes et des milliers, voire des millions de lignes de code, comment vérifier et valider le système dans son intégralité ? Par où commencer ? Comment faire en sorte que toutes les pièces du puzzle s'*imbriquent* parfaitement et *opèrent* à merveille afin que vos parties prenantes (et vos actionnaires) obtiennent ce qu'ils cherchent ? En fait, le processus de qualité commence dès le début de la conception du système.

Les systèmes vastes et complexes sont répartis en plusieurs niveaux, à savoir le système, un ou plusieurs sous-systèmes et ses composants.



Bien qu'il semble logique d'attendre la finalisation du système pour le tester, les bonnes pratiques en vigueur dans le domaine de l'ingénierie système comportent des techniques plus sophistiquées destinées à en assurer la qualité tout le long de son développement. A chaque fois qu'un composant du système (matériel et logiciel) est terminé, il est testé seul, puis est associé aux autres pour former un sous-système. Ensuite, chaque sous-système est testé isolément avant d'être associé aux autres pour former le système. Enfin le système complet sera testé dans son environnement (contexte) afin de vérifier si l'ensemble est fonctionnel en situation réelle.

Outre les tests en cours de conception, certains tests de vérification et de validation peuvent être effectués sur les modèles, par simulation, en vue de détecter les problèmes suffisamment tôt ; en tout cas avant que vous commenciez à réaliser les plisages des métaux et à souder les cartes de circuits imprimés. Vous pouvez même effectuer des vérifications durant la phase d'analyse des exigences afin d'établir si votre interprétation des besoins du client est exacte.

## Test unitaires

Le test du niveau le plus bas du système est étroitement lié à l'implémentation de votre produit. Vous devrez ainsi tester chaque composant matériel et logiciel, résoudre les problèmes éventuels et recommencer l'implémentation. En ce qui concerne le logiciel, cela peut comprendre plusieurs cycles de codage-test-recodage jusqu'au débogage complet.

Après avoir résolu les problèmes détectés, vous pourrez vérifier si le composant répond aux exigences le concernant. Vous vous souvenez du moment où vous avez mis au point la conception détaillée de votre système ? Et bien une partie de cette phase implique la définition d'exigences propres aux composants et la création de tests unitaires associés. Vous devez à présent utiliser les jeux de test définis dans ce plan de tests unitaires pour vérifier si les composants répondent bien aux exigences. Si des défauts sont détectés, revenez sur l'implémentation et trouvez des solutions. Une fois que tous les composants ont été vérifiés fonctionnellement, ils sont prêts à être intégrés aux modules et sous-systèmes des niveaux supérieurs.



Veillez à ce que le programme de tests unitaires comporte une trace des cas de test et des résultats obtenus pour chaque composant. Vous devrez aussi utiliser des matrices de traçabilité pour associer ces tests aux exigences correspondantes. Le système répond à toutes les exigences lorsqu'un résultat positif a été obtenu pour tous les tests.

## *Intégration et vérification des sous-systèmes*

Les composants matériels et logiciels ont été vérifiés unitairement. Ils sont à présent prêts à être intégrés aux modules et sous-systèmes. Si vous avez testé les interfaces, cette étape devrait être relativement rapide.



L'objectif est de vérifier si toutes les interfaces entre les composants et modules ont été correctement implémentées et si les exigences et contraintes du sous-système ont été respectées.

Selon la complexité de votre sous-système, vous devrez peut-être décrire un plan d'intégration définissant l'ordre d'intégration des composants et modules. Planifiez votre intégration de sorte que les éléments inter-connectés puissent être dans la mesure du possible intégrés en même temps. Pour chaque étape d'intégration, testez le fonctionnement des modules en fonction des exigences sur la base du plan de vérification du sous-système que vous avez défini au cours de la phase de conception.



Veillez à prendre en compte les tests réalisés pour vérifier les exigences au niveau des composants, puisque la plupart d'entre elles sont répercutées sur plusieurs niveaux de décomposition. Par exemple, si l'un des critères implique que l'écran devienne noir lorsque l'utilisateur appuie sur un bouton, vous devez vérifier si l'écran peut effectivement être noir (test au niveau du composant). Vous devez aussi vérifier si la pression sur le bouton rend bien l'écran noir (test au niveau du sous-système).

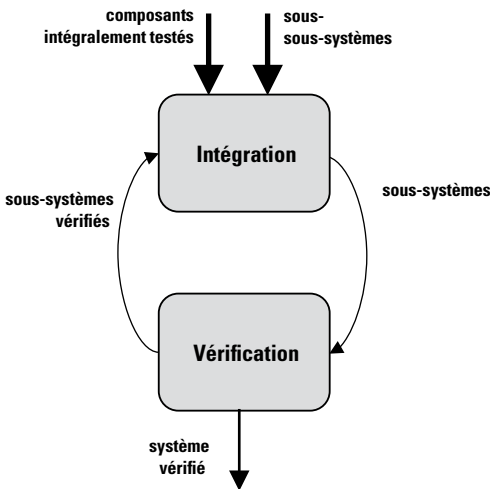
## *Test de système*

Grâce à une itération progressive, parviendrez intégrer, tester et vérifier les sous-systèmes jusqu'à ce que vous atteigniez le niveau du système (v. Figure 5-1). Chaque itération prévoit un test précis et une vérification rigoureuse

du respect des exigences associées. Ce sont d'ailleurs les interfaces qui sont privilégiées. Au niveau supérieur, les tests vous permettront de vérifier si le système global répond aux exigences de haut niveau définies pendant la phase de conception. Une fois de plus, vous devez baser vos tests sur un plan de vérification défini en cohérence avec les exigences du système.



Il est important de documenter les résultats de chaque cas de test et de noter les résultats inattendus et les anomalies. Vous devrez cependant résister à la tentation de résoudre un problème dès son apparition, car vous pourriez perdre le contrôle de la configuration. Au lieu de cela, recensez le problème, analysez les raisons et définissez un plan d'action pour le résoudre dans le cadre d'une approche systématique.



**Figure 5-1** : l'intégration et la vérification sont un processus itératif.

Si tout se passe bien, vous disposerez d'un système vérifié que vous pourrez présenter à vos parties prenantes. Vous pourrez ainsi prouver que toutes les exigences ont été respectées et que le système a été correctement fabriqué.

## Tests d'acceptation du système

« Si tu construis, tout suivra. » Si vous croyez à ce genre de chose, vous risquez d'être déçu. Votre système peut être très bien conçu et implémenté, répondre à toutes les exigences, mais s'il ne remplit pas ses fonctions, il ne sert à rien.

Prenons, par exemple, le cas d'un système de contrôle des feux de signalisation. Conçu pour gérer la séquence des feux dans une grande ville, ce système risque de ne pas atteindre son objectif *principal* (réduire les embouteillages en rationalisant le trafic) si personne n'a pris la peine d'étudier les schémas de circulation pour mieux programmer les cadences des feux.



Le but des tests d'acceptation consiste à vérifier si le système remplit bien son objectif principal.

Pendant la phase conceptuelle de votre projet, vous avez identifié les principaux besoins des parties prenantes, les capacités globales du système, les scénarios d'utilisation (CONOPS et cas d'utilisation) ainsi que les mesures de performance pour la validation du système. Vous avez mis au point un plan de validation du système et si vous avez été malin, vous l'avez rangé dans un coffre afin que personne ne puisse le modifier en vue d'intégrer quelques opérations douteuses permettant de faire de maigres économies. Votre plan de validation du système est la base grâce à laquelle vous pourrez montrer que votre système atteint bien ses objectifs.

Effectuez les tests de validation auprès d'utilisateurs réels et mesurez les performances conformément à votre plan en intégrant si possible une enquête de satisfaction du client. La validation peut impliquer la collecte de données avant, pendant et après le déploiement du système. Après avoir soigneusement documenté les performances du système, rencontrez les parties prenantes, analysez toutes ces données et évaluez la réussite du projet.

Si un problème est détecté pendant les tests, c'est qu'il y a généralement un défaut dans le système qui peut être corrigé en faisant une modification dans la conception. Sachez toutefois que le problème peut être situé ailleurs. Si votre plan de vérification ou votre procédure de test sont erronés ou obsolètes, le test peut échouer sans pour autant que le système présente une défaillance. De même, si une exigence n'est pas correcte, est ambiguë ou incomplète (surtout au niveau de l'interface), c'est vraisemblablement là que se situe le problème. Il faudra donc faire en sorte que les exigences et les plans de test soient les plus exacts possibles.



Lorsque les tests de vérification identifient un problème, revenez en arrière et passez en revue les exigences et la conception. Effectuez les changements nécessaires et recommencez les opérations de test.

Imaginons que vous conceviez un essuie-glace automatique. L'objectif de l'essuie-glace automatique consiste à balayer le pare-brise dès qu'il détecte des gouttes d'eau sur la surface extérieure de celui-ci. L'architecture détaillée de l'essuie-glace automatique comporte un capteur optique associé à une plage d'utilisation spécifique reliée à la surface intérieure du pare-brise, une unité de commande électronique et un logiciel. Votre conception implique l'interaction du capteur et du logiciel en vue de détecter la présence d'eau et l'activation des balais.

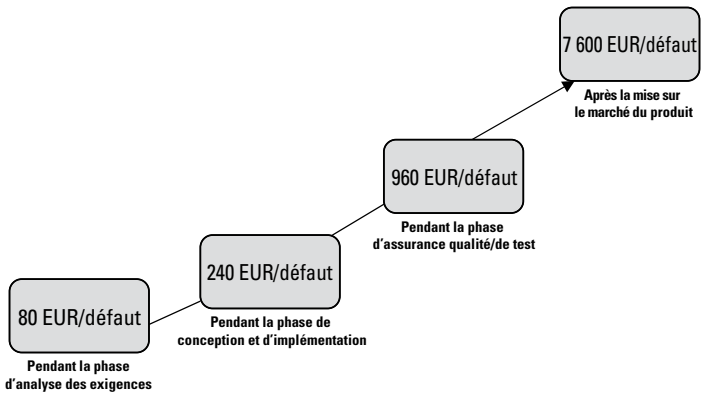
Vous testez chaque composant et vérifiez si l'essuie-glace automatique fonctionne comme prévu dans les limites de la plage d'utilisation du capteur. Ensuite, vous intégrez l'essuie-glace automatique, le pare-brise et les autres sous-systèmes à la voiture. Lorsque vous testez le fonctionnement de l'essuie-glace automatique en présence du système global, il ne fonctionne pas.

Au terme d'une analyse approfondie des causes, vous découvrez la racine du problème : les caractéristiques physiques du pare-brise (précisément, l'index optique et l'épaisseur) ne sont pas compatibles avec la plage d'utilisation du capteur. Vous réalisez que vous n'avez pas spécifié correctement les caractéristiques techniques du pare-brise qui vous assurez la compatibilité avec l'essuie-glace automatique. Vous devez donc revenir en arrière jusque la phase de conception, intégrer l'exigence et retravailler la conception.

Vous avez dû faire face à une *hypothèse* mal appréciée et non vérifiée, l'erreur classique commise par les ingénieurs Systèmes. Cette hypothèse reposait sur le fait que le capteur aurait dû fonctionner avec n'importe quel type de pare-brise. Évidemment, si vous aviez simplement émis cette hypothèse, quelqu'un aurait pu dire : « Pas n'importe quel pare-brise ! ». Vous auriez ainsi compris qu'il vous manquait une exigence. Cette douloureuse expérience montre combien il est important de faire des tests tôt et souvent.

## ***Tester tôt, tester souvent***

Plus tôt vous découvrirez un défaut, moins il vous en coûtera de le résoudre. Comme le montre la Figure 5-2, les défauts constatés après la mise sur le marché du produit coûtent 100 fois plus cher à résoudre que les défauts détectés pendant la phase d'analyse des exigences. Mais ce chiffre n'est qu'un exemple isolé. En adoptant une méthode de test rigoureuse, vous pourrez réduire considérablement les frais nécessaires à la résolution des problèmes.



**Figure 5-2 :** les coûts nécessaires à la résolution des défauts augmentent considérablement au fil du processus de développement.

## Le Diable est dans les . . . interfaces

L'un des plus gros problèmes associés au développement et au test des grands systèmes complexes est dû au fait que les éléments développés séparément ne fonctionnent pas toujours comme prévu lorsqu'ils sont intégrés. Si vous attendez l'implémentation de tous les composants *avant* de les assembler et de les essayer, les problèmes seront détectés probablement trop tard avec un planning et des budgets déjà bien entamés.



L'apparition de problèmes d'intégration tard au cours du processus de développement risque fort d'entraîner le dépassement des délais et des budgets. C'est un facteur de risque *majeur* pour la réussite d'un projet.

Alors ? Que peut-on y faire ? Les ingénieurs Systèmes ont mis au point deux approches pour réduire ce risque :

- Vérifier les interfaces et les interactions entre les principaux sous-systèmes et leurs composants *le plus tôt possible* en utilisant des modèles et en pratiquant des simulations pour un sous-ensemble ou la totalité des sous-systèmes et composants.
- Intégrer les éléments progressivement (de manière itérative) pour voir comment ils fonctionnent au lieu d'attendre qu'ils soient tous prêts pour démarrer leur intégration et détecter les éventuels problèmes.



La première approche, qui consiste à modéliser le système (v. Chapitre 4), peut s'avérer très utile ici. Les modèles d'utilisation présentés sous la forme de diagrammes d'activité et de séquence vous aident à comprendre quels éléments communiquent et quelles interfaces sont nécessaires entre eux. Sur la base de ces données, vous pourrez anticiper l'intégration des éléments interconnectés, réduisant ainsi l'ampleur des travaux de simulation et d'émulation nécessaires. En forçant d'emblée la collaboration des équipes travaillant sur les différents sous-systèmes, les problèmes apparaîtront aussi plus tôt. Vous pourrez ainsi les résoudre plus facilement et à moindre frais.

La modélisation vous permet également d'évaluer votre compréhension des interactions avec les sous-systèmes avant leur réalisation. Vous pouvez ainsi exécuter les modèles et vérifier que ce que vous avez conçu est bien conforme à ce qui est attendu. La transformation des diagrammes en modèles exécutable permet de détecter les défauts au plus tôt, facilitant ainsi leur résolution.

Grâce à des modèles simulant le fonctionnement des composants avant même leur fabrication, vous pouvez tester les interfaces bien avant de graver le système dans l'acier et les circuits imprimés. C'est un peu comme un simulateur de vol permettant aux ingénieurs de tester les différents composants d'un appareil avant qu'il ne quitte le plancher des vaches.

Une deuxième approche consiste à intégrer progressivement les éléments et à les tester (par itérations) le plus tôt possible au cours du processus de développement. Les ingénieurs logiciels utilisent les cycles de développement itératifs depuis des dizaines d'années. Bien sûr, il est plus facile de recourir à l'itération pour le développement de logiciel que dans l'électronique ou l'aéronautique. En effet, les éléments matériels nécessitent plus de temps. Cependant, certains principes conviennent aux deux.

Des cycles d'itération au plus tôt permettent de gérer la complexité des éléments à haut risque en combinant prototypes, matériel existant et logiciel développé spécifiquement. Grâce à ces versions préliminaires exécutable, vous pouvez résoudre les problèmes d'intégration, améliorer les interfaces et vérifier vos choix d'architecture faits en études comparatives.

Certains ingénieurs avant-gardistes conseillent même de tenter d'emblée l'intégration des composants, puis de les tester individuellement en tant qu'éléments du sous-système complet. Bien que cela semble peu intuitif, cela permet aussi de gérer les risques plus tôt au lieu d'attendre que l'intégration mette les problèmes en évidence.



## Chapitre 6

# Aider les grandes équipes à collaborer et à gérer le changement

.....

### *Dans ce chapitre*

- ▶ Chercher des bases communes pour des échanges humains efficaces
  - ▶ Automatiser les processus
  - ▶ Associer les outils et les données relatives au cycle de vie
- .....

**L**es petites équipes de développement de produits savent comment collaborer efficacement : elles savent partager l'information critique, utilisent les mêmes outils de développement, et signalent quand elles modifient une exigence, découvrent un défaut ou organisent une fête !

Multipliez la taille de l'équipe par 100, donnez-leur une liste de 700 exigences souhaitées. Dites-leur que vous voulez le produit dans six mois et vous n'aurez plus qu'à chercher un autre boulot (en plus, vous ne risquez plus d'être invité de sitôt aux soirées entre collègues).

Comment adapter la magie des petites équipes aux grandes ? Vous allez l'apprendre dans ce chapitre.

## *Les enseignements de Facebook*

Le plus célèbre réseau social a beaucoup de choses à nous apprendre. Créé en 2004 alors que la plupart des gens équipés disposaient déjà d'une adresse électronique, Facebook a été d'emblée une réussite. Il comptait d'ailleurs 600 millions d'utilisateurs en 2011. Mais qu'est-ce qui fait la popularité de ce réseau social ?

Les fondateurs de Facebook ont trouvé un moyen de mettre au point une plate-forme conviviale et performante, organisée autour des principales interactions sociales impliquant amis, famille et autres personnes affichant des intérêts communs. Son interface est très facile d'utilisation, permettant de partager des informations et des photos, d'exprimer ce que vous aimez ou non et bien d'autres choses encore. Au lieu de forcer les gens à communiquer dans un format complexe, comme c'est le cas pour les courriers électroniques, Facebook utilise la technologie comme une plate-forme favorisant les échanges.

Imaginez à présent appliquer la « révolution Facebook » à l'univers de l'ingénierie système. Imaginez une plate-forme technologique facilitant les interactions entre les équipes de développement, les ingénieurs et parties prenantes. À l'instar de Facebook, celle-ci exploiterait l'Internet pour rapprocher les gens, rendant une équipe dispersée aussi efficace qu'un groupe travaillant dans un même bureau.

## *Mettre tout le monde au diapason*

Il faut bien plus qu'une ingénierie brillante pour créer un produit intelligent emportant les suffrages du marché. Les recherches ont montré qu'un tiers des produits fabriqués ne respectaient pas les critères de performances ou les exigences. Par ailleurs, 24 % des projets sont annulés en raison de retards irrécupérables. Bien souvent, la racine du problème n'est pas dans la conception du système, mais dans le manque de connaissances et de communication.

Ce n'est pas un hasard si les gros projets font les frais des erreurs de communication. La plupart des grandes équipes de développement sont réparties sur plusieurs villes, entreprises et pays. Les barrières linguistiques et culturelles rendent les échanges difficiles tandis que le décalage horaire handicape souvent la collaboration. Même pour les employés d'une même entreprise, les structures en silo entravent la communication, limitent la productivité et propagent la tendance à blâmer autrui.

Les problèmes de communication peuvent entraîner notamment :

- ✓ une définition floue des objectifs du système,
- ✓ des interprétations multiples des exigences système,
- ✓ des exigences incomplètes ou négligées,
- ✓ du temps perdu à rassembler les données manuellement à partir de plusieurs sources,
- ✓ des équipes travaillant avec des documents obsolètes,
- ✓ des responsabilités oubliées ou redondantes.

Et comme si cela ne suffisait pas, les équipes de développement subissent aussi une pression énorme pour augmenter leur productivité, alors même que les systèmes deviennent de plus en plus complexes. Par ailleurs, des produits intelligents à fort contenu logiciel exigent davantage de documentation. La courbe d'apprentissage est donc plutôt rude pour les nouvelles recrues.



Le meilleur moyen de venir à bout de ces difficultés propres aux grandes équipes consiste à définir une base commune pour le développement et la maintenance. Un langage standard est également nécessaire pour stimuler les échanges.

## *Jeter les bases de la collaboration*

La plupart des systèmes intelligents modernes contiennent de multiples sous-systèmes provenant de sources variées ainsi que des millions de lignes de code développées par des équipes d'ingénieurs travaillant dans des entreprises, pays et cultures différents. Prenons l'exemple d'un avion, la cellule est généralement fabriquée par telle entreprise de tel pays, le moteur par une autre, l'avionique par une troisième et le logiciel d'intégration par une autre encore. Pour simplifier les processus de développement et de test, il est essentiel de fournir une plate-forme commune de développement du système.

L'utilisation d'une plate-forme commune permet de briser les barrières entre les équipes. Les ingénieurs peuvent ainsi collaborer tout au long du cycle de développement. Une plate-forme unifiée facilite la mise en commun des travaux et le partage des connaissances ; ce qui permet de gagner du temps sur le cycle de développement. En réduisant les problèmes de communication

et en simplifiant les processus, la qualité et la satisfaction de l'équipe ne peuvent que s'améliorer. Par ailleurs, les ingénieurs pourront tenir l'équipe informée de l'évolution du projet grâce à des tableaux de bord qui aideront les chefs de projet à garder les développements sur la bonne voie.

## *Parler la même langue*

Pour rassembler des équipes différentes par leur culture et leur discipline, rien ne vaut l'adoption d'une approche basée sur les modèles privilégiant un langage commun, quel que soit le domaine. En proposant une référence visuelle pour la conception du système, la modélisation permet de venir à bout de la barrière de la langue. Chacun peut ainsi mieux appréhender le système et mettre ses connaissances multi-domaines à la disposition de l'équipe. Une compréhension commune se transforme directement en amélioration de la productivité, les ingénieurs ne perdant plus leur temps à résoudre des problèmes venant d'une mauvaise compréhension.

Le langage SysML devient la norme pour la modélisation des systèmes. Il adresse les différentes activités de développement incluant la définition des exigences, l'analyse, la conception, la vérification et la validation des systèmes constitués de matériel, logiciel, données, personnel, voire même d'installations. Il présente également l'avantage d'être totalement compatible avec le langage UML, ce qui facilite considérablement le passage du système au logiciel.

Il existe de nombreux outils logiciels de modélisation SysML. Chacun d'entre eux dispose de son propre environnement de développement. Pour permettre une collaboration plus efficace, les équipes doivent opter pour une plate-forme de travail standard dotée des outils les plus efficaces du marché.

## *Au-delà de la messagerie électronique et du partage de documents*

Mettre en place une plate-forme de développement commune représente une étape majeure pour un travail d'équipe plus efficace. Mais cela ne suffit pas. Si l'un des membres de l'équipe de développement apporte un changement (à une exigence ou à un élément de modélisation par exemple) et que celui-ci n'est pas immédiatement signalé au reste de l'équipe, cela peut provoquer un véritable chaos.

## Assurer le suivi des changements

Lors de la conception d'un système, les données critiques évoluent forcément en permanence. L'ingénierie système implique notamment des processus itératifs pour la conception et le test: vous concevez votre système, vous développez des modèles, vous les testez, vous retravaillez les modèles d'origine pour résoudre les problèmes et ainsi de suite. Vous adaptez donc en permanence les modèles, les résultats des tests et les autres données du projet. Les exigences peuvent évoluer elles aussi, de même que le marché et l'entreprise.

Traditionnellement, les grandes équipes se basent sur des documents textuels contenant toutes les données du projet. Des armoires remplies de dossiers reprenant les exigences, les données structurelles, celles propres à la modélisation et bien d'autres encore représentent la base des connaissances du système. Mais les documents sont restrictifs car ils ne font que *recenser* les informations sans faciliter leur évolution. Si vous ne vous basez que sur ces documents, sachez qu'ils seront dépassés avant même d'avoir été approuvés !



Les équipes de développement ont besoin de mécanismes flexibles et cohérents permettant d'enregistrer les données critiques tout en simplifiant leurs modifications mais dans le respect de leur intégrité.

## Signaler les changements

La méthode traditionnelle qui consiste à créer plusieurs dizaines de documents stratégiques et d'échanger les données par courriel ne fonctionne pas très bien dans les environnements de développement tels que nous les connaissons aujourd'hui. Les exigences propres à un système complexe se comptent par centaines, voire par milliers. Les seuls échanges par courriel ne feraient que plonger l'équipe dans le chaos.



Les équipes de développement ont donc besoin d'un outil permettant des échanges efficaces entre leurs membres, où qu'ils soient. Une plate-forme robuste représente le seul moyen d'éviter les problèmes qui apparaissent lorsque l'on ne sait plus très bien quelle est la dernière version d'un document.

## Décider de ce qu'il convient de partager

Si vous deviez dresser la liste de toutes les informations critiques qui doivent être prises en compte pour le développement d'un système complexe, vous risqueriez de ne jamais en venir à bout. Lorsque vous sélectionnez les données devant être transmises à chaque membre de votre équipe, veillez à éviter de dresser la liste exhaustive des informations échangées.



Votre objectif est de faciliter le partage et la collaboration entre les membres d'une équipe de développement. Il ne s'agit pas de les accabler de données inutiles. Établissez donc la liste des informations essentielles qui doivent être communiquées afin de stimuler la collaboration, à savoir :

- ✓ Priorités en matière de développement
- ✓ Approbations pour le projet
- ✓ Plannings
- ✓ Rôles et responsabilités des employés
- ✓ Exigences
- ✓ Demandes de modification
- ✓ Modèles conceptuels
- ✓ Cas d'utilisation
- ✓ Plans de test
- ✓ Défauts
- ✓ Problèmes critiques
- ✓ Données de traçabilité
- ✓ Budget
- ✓ Informations sur l'approvisionnement

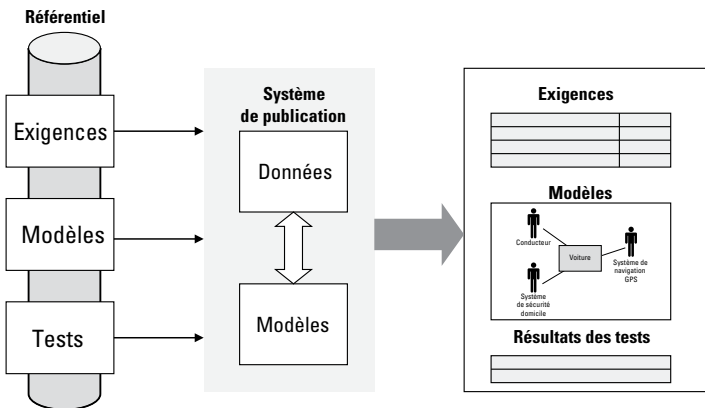
# Explorer les méthodes propices au partage

On ne peut s’attendre à ce que les centaines d’entreprises qui participent au développement d’un produit utilisent les mêmes outils qui, de surcroît, proviendraient d’un même fournisseur. Dès lors, il faut donner les moyens à l’équipe de mettre ses données en commun, quelle que soit leur origine.



Une collaboration efficace commence par une plate-forme permettant la gestion et le suivi globaux du processus de développement. Grâce à cette plate-forme, les parties prenantes et les ingénieurs produisent et traitent des données qui sont partagées, analysées et utilisées dans de la documentation tout au long du cycle de développement des systèmes.

Les outils d’automatisation et de fédération utilisant de la technologie pour faciliter la communication et automatiser les processus de travail représentent l’étape suivante pour une collaboration généralisée.



**Figure 6-1** : un référentiel virtuel contenant toutes les données du projet permet de renforcer la collaboration.

## Créer un référentiel virtuel contenant des données actualisées

Les participants doivent accéder en temps réel aux informations actualisées du projet, et ce, quel que soit l'endroit où elles sont stockées et quel que soit leur format. Les ingénieurs Systèmes consacrent généralement beaucoup de temps à assurer le suivi des données et veiller à ce que chaque participant impliqué travaille avec les bons documents.



En organisant les données critiques dans un référentiel virtuel, vous pourrez créer une base de connaissance unique contenant les éléments essentiels du développement. Vous pourrez aussi automatiser le partage des informations.

Le référentiel *virtuel* n'existe pas physiquement. Les données sont récupérées en fonction des besoins à partir des sites répartis partout dans le monde. Ainsi, les données propres à l'ingénierie mécanique pourront être récupérées d'un référentiel physique géré par l'équipe compétente à Paris tandis que les informations relatives à l'ingénierie électrique proviendront d'une base de données à Tokyo. Les utilisateurs doivent savoir où se trouvent les données physiques, dans la mesure où ils doivent obtenir ce dont ils ont besoin quand ils le veulent.

La Figure 6-1 présente un modèle conceptuel de ce type de référentiel virtuel et son utilisation. Le chef de projet est en quelque sorte le cerveau des opérations. Il gère l'exécution des activités de développement. Il peut accéder aux données en fonction des besoins grâce à des normes ouvertes. Au lieu de transmettre des documents textuels de proche en proche, des chefs de produit aux architectes, des concepteurs aux testeurs, etc., chaque membre concerné peut accéder aux données en toute transparence.

Un référentiel virtuel tel que celui-là représente une source unique de données actualisées auxquelles chacun peut accéder où qu'il se trouve. De plus, les utilisateurs peuvent s'échanger des informations, partager des idées, prendre connaissance des décisions et participer aux réflexions, un peu comme s'ils travaillaient dans le même bureau.



## Qu'est-ce qu'un référentiel « virtuel » ?

Pourquoi privilégier le virtuel au réel ? Les raisons poussant à ne pas « tout stocker dans une même base de données » sont nombreuses.

- ✓ Tout d'abord, si vous tentez de créer une « base de données universelle » capable de stocker tout ce dont vous pourriez avoir besoin, sachez que cela représente un travail pharaonique !
- ✓ Ensuite, vous devrez pouvoir assurer la compatibilité avec chaque outil disponible sur le marché aujourd'hui comme demain.
- ✓ Enfin, une base de données volumineuse autorisant un accès partout dans le monde sera très réactive pour certains, mais aussi

très lente pour d'autres ; sans parler des problèmes associés aux mises à jour, les pannes (réseau ou matérielles) et les nombreux autres problèmes associés au stockage des données sur un support unique.

L'approche virtualisée est bien meilleure : elle permet la distribution et l'optimisation des données quel que soit l'outil utilisé, ainsi que des performances supérieures, une mise à niveau et une connexion bien plus facile.

**Attention !** Si vous optez pour une base de données regroupant tout, vous risquez de vous confiner à un seul fournisseur. Vous limitez donc vos choix en tant que consommateur.

## Simplifier l'intégration des outils

Bien que le partage des ressources et des actifs par le biais d'un référentiel virtuel semble évident, ce n'est pourtant pas si simple. Les outils internes, les projets open source et les nombreux fournisseurs peuvent représenter des barrières en raison de l'incompatibilité des formats de données notamment.

Au cours de ces dernières années, les fournisseurs des outils de développement les plus populaires (v. Tableau 6-1) se sont rassemblés en vue d'établir des normes pour l'intégration des outils. Ils ont formé la communauté OSLC (Open Services for Lifecycle Collaboration) et se sont engagés à promouvoir de nouvelles formes de collaboration en éliminant les barrières entre outils.

OSLC s'inspire de l'architecture Internet pour définir un ensemble de normes cohérentes entre elles, des formats de ressources communs et des services pour le partage de celles-ci. OSLC permet d'intégrer plus facilement les outils en provenance de divers fournisseurs par le partage des données du cycle de vie des systèmes, à savoir les exigences, les demandes de modification, les cas de test et les défauts.

**Tableau 6-1 Principaux outils d'ingénierie système**

<i>Type d'outil</i>	<i>Compétences</i>
Gestion des exigences et traçabilité	Traçabilité de bout-en-bout des exigences sources, de mission et de système / sous-système
Développement de systèmes guidé par la modélisation	Modélisation des exigences et des fonctions du système, réalisation, études comparatives, simulation et validation
Gestion des changements et de la configuration	Gestion de la collaboration, des changements, du référentiel partagé et de la configuration
Production automatique des documents	Production des documents contenant les exigences, les modèles et les données techniques
Ingénierie intégrée des systèmes et du logiciel	Itérations sur les exigences et modèles, développement intégré

## *Automatisation de la production de document*

Même dans un développement basé sur la modélisation, il est nécessaire de produire des documents et rapports pour des raisons contractuelles, de conformité, d'analyse technique ou de gestion de projet. La documentation implique souvent un gros travail manuel : il faudra, par exemple, extraire un graphique à partir d'un outil de modélisation en réalisant une capture d'écran pour ensuite l'intégrer dans tel ou tel document. De plus, les ingénieurs consacrent énormément de temps à collecter les informations venant de diverses sources. Ils doivent vérifier qu'ils disposent bien des dernières données, les synthétiser et les remettre en forme pour produire un rapport personnalisé.



Les outils d'automatisation des documents facilitent la production de rapports personnalisés, tout en assurant la cohérence des informations extraites.

Les outils d'automatisation vous permettent d'accéder à un référentiel centralisé, d'identifier les types d'information dont vous avez besoin, les sélectionner et produire le rapport. Le processus en est simplifié, mais ils apportent d'autres bénéfices comme la réutilisation et la cohérence des données ; vous pouvez rassembler les données critiques en provenance de plusieurs sources, même plusieurs fournisseurs, dans un seul et même rapport.

Cela permet donc de gagner du temps lors de la production des documents, mais cette solution exprime tout son potentiel lorsqu'un changement arrive. Il vous suffit d'apporter les modifications souhaitées aux exigences et aux modèles. Vous n'aurez ensuite qu'à appuyer sur un bouton pour consulter les documents mis à jour.

Ces documents sont protégés par le copyright © 2012 John Wiley & Sons, Inc. Toute divulgation, toute distribution et tout usage non autorisés sont strictement interdits.

## Chapitre 7

# Dix manières de réussir son Ingénierie Système

### *Dans ce chapitre*

- ▶ Résoudre les défauts de conception dès le début du processus de développement
- ▶ Développer des modèles stratégiques flexibles
- ▶ Maîtriser le développement de logiciel intégré complexe
- ▶ Améliorer la gestion des exigences
- ▶ Réutiliser le code pour accélérer l'implémentation du produit
- ▶ Utiliser des outils de collaboration
- ▶ Respecter les délais de mise sur le marché des solutions complexes
- ▶ Utiliser une plate-forme de développement intégré
- ▶ Tester tôt, tester souvent
- ▶ Mettre les exigences en commun pour gagner en efficacité

**L'**ingénierie système vous donne un avantage compétitif lorsque vous développez des produits intelligents, elle apporte une réelle plus-value à vos clients (et à votre chiffre d'affaires). En mettant l'ingénierie système au cœur de vos activités, vous parviendrez à réaliser des produits attractifs. Vous éviterez les erreurs coûteuses et renforcerez votre capacité d'adaptation à la dynamique du marché.

Dans ce chapitre, vous passerez en revue dix exemples d'entreprise qui ont adopté les meilleures pratiques en matière d'ingénierie système, obtenant ainsi des résultats réels et quantifiables.

## Résoudre les défauts de conception avant qu'ils ne fassent la une des réseaux sociaux



L'identification précoce des défauts constitue l'un des principaux défis lors de la conception de produits intelligents. La plupart des défauts sont introduits lors des phases de conception. Ils ne sont cependant pas détectés avant la phase de test, voire de production.

Pour les produits de diffusion de masse, le fait de passer à côté de tels défauts peut s'avérer extrêmement coûteux. Mais pour les systèmes de défense, cela peut être en plus très dangereux. Pour compliquer davantage les choses, la plupart des systèmes de défense se composent de plusieurs sous-systèmes complexes conçus par un ensemble de sous-traitants agréés.

Brockwell Technologies est l'un des partenaires du Département de la Défense américain. L'entreprise basée à Huntsville en Alabama fabrique des systèmes d'armes embarqués et réalise des tests de diagnostic destinés aux véhicules militaires. La réalisation de systèmes d'armes interconnectés comporte un véritable défi : comment modifier un système sans que les autres en soient affectés ?

Pour limiter le risque d'introduire des défauts d'interactions avec les sous-systèmes, Brockwell Technologies a intégré dans son processus de développement des techniques de modélisation pour la conception et les tests. En modélisant la structure et le comportement du système, les ingénieurs de Brockwell sont parvenus à réaliser des prototypes d'ensembles complexes afin de visualiser leur fonctionnement. Cette technique de modélisation prédictive permet aux ingénieurs d'identifier les failles très tôt lors du processus de développement, bien avant que les systèmes soient fabriqués en série.

Brockwell a investi dans l'ingénierie système. Tant l'entreprise que le Département de la Défense en ont tiré profit, Brockwell propose désormais des systèmes d'armement plus robustes et plus sûrs. De plus, les délais de commercialisation ont été réduits de 40 %.

## ***Concevoir un modèle commercial flexible***

Si vous comptez lancer une nouvelle activité ou accéder à un nouveau marché, le moyen le plus rapide consiste à mettre en place une infrastructure commerciale polyvalente destinée à devenir l'image même de l'ingénierie système. C'est exactement ce qu'a fait une entreprise d'Atlanta lorsqu'elle est entrée dans le marché très concurrentielle de la télématique il y a quelques années.

L'entreprise a compris qu'elle pouvait laisser ses concurrents sur le carreau en renforçant son infrastructure commerciale de telle sorte à adapter rapidement ses relations commerciales et ses modèles de fourniture de services. Grâce à un ensemble de processus basé sur une technologie ouverte, l'entreprise a pu saisir de nouvelles occasions devant des sociétés bien établies mais engoncées dans leurs technologies propriétaires et leurs modèles commerciaux rigides.

L'entreprise a conçu tous ses principaux processus (ventes, administration, opérations) avec un seul objectif en tête : lancer rapidement de nouveaux services. Par ailleurs, pour faciliter leur développement, l'entreprise a installé une nouvelle plate-forme commune d'ingénierie système collaboratifs. Son infrastructure de développement lui a permis de gérer ses produits et fournitures, de les intégrer dans un référentiel unique, de faciliter la collaboration entre des équipes réparties à l'échelle mondiale, d'assurer la gestion des versions et de signaler rapidement les changements.

Grâce à une combinaison de systèmes ouverts, de processus flexibles et une collaboration plus efficace, l'entreprise pouvait mettre de nouveaux services sur le marché en moins de 30 jours. Les tendances dans la télématique évoluent, délaissant la technologie pure pour privilégier les services novateurs exploitant les données des véhicules de manière toujours plus originales. L'entreprise est donc plutôt bien placée dans la course au succès.

## ***Garder le contrôle des logiciels intégrés complexes***

Les entreprises performantes savent que, quand leur produit fait partie d'un ensemble plus complexe composé de système de systèmes, elles doivent veiller

à ce que la qualité du produit occupe la place centrale. Après tout, vous ne souhaitez pas que votre société devienne le maillon faible de la chaîne. La taille et la complexité des logiciels ne cessant d'augmenter, il est de plus en plus difficile de garantir la qualité.

Pendant longtemps, le développement de logiciel « à la main » a été l'activité principale d'un fournisseur de services technologiques allemand spécialisé dans les systèmes de mesure et de contrôle ainsi que dans l'ingénierie des procédés. Lorsque l'entreprise a commencé à mettre au point des logiciels intégrés complexes destinés aux systèmes contrôlant à distance les installations photovoltaïques, elle a compris qu'elle devait opter pour un nouvel outillage logiciel.

Elle s'est fixée quatre objectifs : réduire les défauts des produits, améliorer la traçabilité, favoriser la réutilisation des modules logiciels et assurer une qualité homogène des produits. L'entreprise a sélectionné une plate-forme basée sur la modélisation favorisant l'ingénierie système temps-réel et embarqués. La nouvelle plate-forme permet à l'entreprise d'identifier et réparer les problèmes au plus tôt grâce aux tests effectués à l'aide de modèles de conception. Pour cela, elle utilise un logiciel de grande qualité. Bénéfice additionnel, l'entreprise peut désormais produire des modules logiciels et des sous-systèmes réutilisables. Elle jouit ainsi d'un avantage considérable sur ses concurrents.

## *Améliorer l'efficacité grâce à des exigences cohérentes*

Pour fabriquer le bon produit et le faire correctement, vous devez bien comprendre les besoins des clients et des parties prenantes. Vous devez aussi spécifier les exigences de votre système à partir de ces besoins. Sans une gestion efficace des exigences, vous risquez de facilement perdre de vue vos objectifs. Et vos clients ne vous attendront pas.

Face à des centaines de développeurs dispersés dans le monde entier, qui utilisaient tous des outils différents de gestion d'exigences, une entreprise australienne rencontrait de nombreux problèmes, notamment en ce qui concernait la traçabilité. La direction s'inquiétait car les exigences ne pouvaient pas être testées de manière uniforme étant donné que chaque équipe de développement avait adopté sa propre méthode de gestion. Pis encore : le manque de cohérence laissait une grande place à l'erreur, ce qui risquait de compromettre les relations de l'entreprise avec son plus gros client, qui n'était autre que la Défense australienne.

Elle a donc mis en place un système global pour la gestion des exigences unique à travers toute l'entreprise. En favorisant l'accès à un référentiel centralisant toutes les exigences, ils ont pu éliminer la confusion, les remaniements coûteux et les duplications d'effort. Plus important encore, cette solution favorisait la traçabilité des exigences, le produit fini pouvait enfin répondre complètement aux exigences du client.

## *Réutiliser le code pour accélérer les lancements de produit*

Si vous comptez dans votre portefeuille une gamme de produits dont les fonctions principales sont proches, il est fort probable qu'une partie du code soit similaire. Les entreprises intelligentes structurent leur code en modules réutilisables. Elles peuvent ainsi accélérer le développement des produits futurs.

C'est exactement ce qu'Océ N.V. avait en tête lorsqu'elle a lancé l'imprimante pour feuilles coupées la plus rapide au monde. Océ occupe une position dominante sur le marché des technologies et services de gestion numérique des documents. L'entreprise développe des applications logicielles avancées permettant le transfert des documents et des données sur les réseaux internes et par Internet vers les périphériques d'impression et d'archivage en local et dans le monde entier.

En règle générale, Océ programme chaque nouvelle imprimante en repartant de zéro, mais lorsqu'elle a dû gérer le code pour les 17 processeurs composant son nouveau modèle, elle a revu son approche.

Océ a déployé des outils de développement guidé par la modélisation afin de décomposer le système d'impression en plusieurs sous-systèmes plus faciles à gérer. Cela a permis aux développeurs de modéliser une série d'automates concurrents qui ont ensuite été codés sous la forme de modules réutilisables pour des environnements cibles variés. Désormais, dès qu'une modification s'avère nécessaire, Océ met à jour les modèles et régénère le code, ce qui lui permet de réduire considérablement les délais de gestion des changements.



Aujourd'hui, plus de 50 % du code de l'entreprise est réutilisable. Elle profite ainsi d'améliorations considérables en termes d'efficacité et de qualité. En fait, Océ a pu mettre sur le marché un prototype fonctionnel en deux mois seulement, soit six mois plus tôt que prévu.

## *Rationaliser le développement grâce aux outils de collaboration*

Les entreprises qui développent des produits complexes, intelligents et super-équipés savent que la réussite dépend autant de leurs expertises techniques que de leurs capacités à mener une ingénierie de qualité. Si une certaine discipline n'est pas mise en place pour coordonner les différents groupes travaillant sur le projet, les produits complexes seront probablement à l'échec.

Lorsque General Motors (GM) a travaillé sur la Chevrolet Volt, elle a déployé des efforts considérables pour établir les meilleures pratiques en termes d'ingénierie système. GM a passé en revue ses méthodes de développement et de gestion des travaux techniques dans le but de les améliorer.

Auparavant, les ingénieurs passaient la plupart de leur temps à surveiller le travail des développeurs en veillant à ce que chacun dispose de la même version des exigences et des documents nécessaires. Pour que les ingénieurs puissent davantage se concentrer sur la fonctionnalité et la qualité, GM a décidé de mettre en place un outil commercial afin de coordonner les équipes de développement et ne maintenir qu'une seule version publique des exigences et documents.

GM a ensuite déployé des pratiques de développement guidé par la modélisation pour une meilleure maîtrise de la complexité (la Volt exécute environ 10 millions de lignes de code). Les développeurs de GM ont construit des modèles pour visualiser les interactions entre les systèmes intégrés et des simulations pour tester ces modèles.

L'association d'outils de collaboration et de méthodes de développement guidé par la modélisation s'est avérée une véritable réussite. GM a développé la Volt en 29 mois seulement ; un record pour GM étant donné qu'en règle générale, la mise au point d'un véhicule prend au moins cinq ans.



## *Mettre des solutions complexes sur le marché et dans les temps*

Pour réussir sur un marché où la concurrence fait rage, les entreprises qui développent des grands systèmes complexes savent qu'elles doivent opter pour une approche plus intelligente des exigences lors du développement de leurs produits. La capacité à appréhender et gérer ces exigences peut faire la différence entre l'obtention et la perte d'un gros contrat.

Une société active dans le secteur de la défense voulait réduire les risques associés à la mise sur le marché de systèmes de systèmes dont la valeur s'élevait à plusieurs millions de dollars. Elle a donc eu l'idée d'intégrer la gestion des exigences à son infrastructure de développement de manière à ce que l'entreprise soit capable de mettre sur le marché rapidement des systèmes complexes respectant ces exigences.

Elle a basé son approche sur des produits disponibles sur le marché pour la gestion des exigences et la planification de l'architecture de système. À présent, l'entreprise peut fournir non seulement des solutions technologiques, mais aussi opérationnelles, techniques, pédagogiques et systémiques. Elle parvient ainsi à mettre sur le marché des grands systèmes complexes de qualité plus rapidement que jamais. De plus, en cas de modification des exigences du client, l'entreprise peut très rapidement s'adapter.

## *Renforcer la productivité grâce à une plate-forme de développement intégré*

Les entreprises qui mettent au point des systèmes critiques doivent montrer que leur processus de développement respecte les normes de sécurité nationales et internationales. Si leur environnement est fragmenté, elles peineront à démontrer cette conformité.

En raison de la diversité des outils de développement utilisés et de la répartition des équipes de développement sur deux principaux sites, une autre entreprise australienne a dû s'adapter pour garantir la réussite des projets à venir. Cette société est spécialisée dans les systèmes de commande et de signalisation pour les chemins de fer du monde entier. Ses clients exigent donc des produits qui soient compatibles avec leurs installations.

L'entreprise a donc dû mettre en place un environnement assurant le respect des normes de sécurité et de fiabilité en vigueur. Pour cela, elle s'est appuyée sur une solution de gestion des exigences et de la configuration capable de répondre aux besoins des clients, mais aussi de stimuler la collaboration entre les équipes de développement.

## *Tester tôt, tester souvent*

Le fait de pouvoir mettre un modèle exécutable ou un prototype dans les mains des parties prenantes constitue souvent un véritable atout. Si vous obtenez un avis suffisamment tôt sur votre travail, vous gagnerez du temps par la suite.



Votre capacité à effectuer des tests simples le plus tôt possible constitue également un atout majeur pour les tests finaux. En laissant les équipes de test accéder rapidement à vos produits, même si les testeurs savent pertinemment que le produit n'est pas complet ou qu'il ne s'agit que d'un prototype, vous améliorerez vos procédures de test et permettrez aux concepteurs de mettre leurs idées à l'épreuve.

## *Partager les exigences pour gagner du temps et de l'argent*

Les grandes entreprises de développement comptent souvent plusieurs équipes dispersées qui travaillent sur des versions parallèles, sur la base d'exigences communes. Elles pourraient gagner énormément de temps si elles avaient à leur disposition un mécanisme de partage des exigences.

Une société du Michigan (États-Unis) occupe une position dominante sur le marché de l'électronique mobile et des systèmes de transport. Elle avait pour objectif d'améliorer la communication entre ses équipes afin d'améliorer leur productivité lorsque celles-ci développaient des versions parallèles sur la base d'exigences communes.

La mise en place d'un outil de gestion des besoins a permis aux développeurs de partager les données. Ils pouvaient ainsi importer les exigences à partir d'un référentiel en vue de les réutiliser dans un nouveau projet, évitant ainsi toute duplication d'effort. Parmi les autres avantages de cette nouvelle approche : un gain de temps et une diminution des coûts de développement. Étant donné que l'outil de gestion des exigences améliore la cohérence et stimule la collaboration, l'entreprise est capable de proposer des produits de meilleure qualité dans un délai bien plus court.