

## Exigences fonctionnelles et non fonctionnelles

Lorsque l'on cherche à développer un logiciel ou un système, la première chose à faire est de récolter les exigences. On distingue généralement deux types d'exigences :

- les exigences dites **fonctionnelles**, liées à la description de ce que le système livré doit être capable de faire (i.e. ses fonctions).
- Les exigences dites **non fonctionnelles**, spécifiant quelque chose sur le système lui-même et la manière dont il exécute ses fonctions.

Il existe de nombreuses sortes d'exigences non fonctionnelles. La norme ISO-IEC 25010 distingue les exigences non fonctionnelles relatives au produit lui-même (*product quality*), et celles relatives à son usage (*quality in use*). La norme fixe les définitions et permet d'avoir une vision unifiée des attributs de la qualité.

Product Quality							
Functional Suitability	Reliability	Performance Efficiency	Usability	Maintainability	Security	Compatibility	Portability
Functional completeness	Maturity	Time behaviour	Appropriateness recognisability	Modularity	Confidentiality	Co-existence	Adaptability
Functional correctness	Availability	Resource utilization	Learnability	Reusability	Integrity	Interoperability	Installability
Functional appropriateness	Fault tolerance	Capacity	Operability	Analysability	Non-repudiation		Replaceability
	Recoverability		User error protection	Modifiability	Accountability		
			User interface aesthetics	Testability	Authenticity		
			Accessibility				

Quality in use				
Satisfaction	Effectiveness	Freedom for risk	Efficiency	Context Coverage
Usefulness		Economic risk mitigation		Context completeness
Trust		Health and safety risk mitigation		Flexibility
Pleasure		Environmental risk mitigation		
Comfort				

L'architecture d'un système **influence de façon significative** l'accomplissement de certaines propriétés non fonctionnelles. C'est pourquoi, l'architecte doit tenir comptes des exigences non fonctionnelles quand il élabore une architecture.