

Use Case Diagram



Español



Entries

- Requirements Specification Document
 - ◆ Can it be used configuration management?: **Yes**



Exit

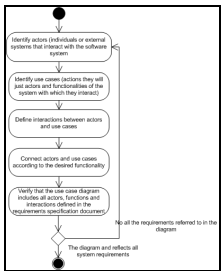
- Use Case Diagram (usingUML)
 - ◆ Can it be used configuration management?: **Yes**



Solution



Process



Development time

- ◆ To acquire the necessary knowledge to develop the software product: 1 day
- ◆ To create the Product Pattern : 1 day
- ◆ To apply the Product Pattern: 1 day



Explanation video

- ◆ Not applicable



Related Patterns

- High Level Use Case Diagram
- Use Case Diagram in Expanded Format
- Requirements Specification



Quality Controllers

- None



Templates

- UML notation Use Cases
- Template to document the use cases
- Template use cases diagram



Examples

- Example use case "Login", "LogOut" and "ModifyData" in the context of a social network.



Support Tools

- Argo UML (Open Source Project)
- Dia (GNOME)
- Rational Software Modeler (IBM)
- StarUML (Open Source Project)
- Visual Paradigm for UML (Visual Paradigm)



Initial Context

It takes into account the requirements specification document, and taking as a starting point, it desired to create a global representation showing the functioning of a software system; it desired to include in such representation the interactions between the software system to develop and its actors (either people or external systems).



Result Context

It takes into account with a UML-based standard that represents the full functionality of a software system satisfying the requirements SpecificationLMS, detailing how the system can be used and by identifying the interactions between the software system and its actors.



Problem

It desired to create a representation of functioning of a software system from the user's perspective. Within this representation is desired detail the different types of users and external systems that interact with the software system to be developed. The development group already has the requirements specification document.



Restrictions (*Forces*)

- **Type of Organization:** SMEs, Large Companies.
- **System Type:** It applies to all types of systems.
- **Programming Paradigm:** OO (Object Oriented)



Roles

- Analyst
- Customer
- Project Manager
- System Users



Lessons Learned

- Benefits of using this pattern
 - ◆ A graphic description of how the software system works and how it will be used is provided.
 - ◆ It is possible to describe the interactions between the system and its actors.
 - ◆ It has a representation that is useful to show the client what will be the system's functionality and roles are involved.



Capability Level

- Capability Level 2 CMMI.



Basic Knowledge and Skills



Knowledge

- Definition of software requirements (functional and non-functional).
- Notions of UML, especially the technique of modeling of use cases.



Abilities

- Capacity for abstraction.



Information Resources

- Amescua A., et al. (2003). *Análisis y Diseño Estructurado y Orientado a Objetos del Sistema Informáticos*. McGraw Hill/Interamericana de España, S.A.U.
- Ferré Grau, X & Sanchez-Segura, M. (2004). *Desarrollo Orientado a Objetos con UML*. Recuperado el 2009-11-26 de <http://www.clikear.com/manuales/uml/index.aspx>.
- Fowler, M. (2004). *UML distilled: a brief guide to the standard object modelling language*. Addison-Wesley.
- Jacobson, I. & Booch, G.(1999). *The unified software development process*. Addison-Wesley.
- Rumbaugh, J. & Jacobson, I. (2005). *The unified modeling language reference manual*. Addison-Wesley.