

Introducing Use Cases in a small Organization: An Experience and Lessons Learned

Dolors Costal, Xavier Franch, Nuria Rodríguez,
Luis Delgado and Carme Jiménez



Group of Software Engineering for Information Systems
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Outline

- The PRISMA organization
- Departing situation
- Decisions made
- Lessons learned
- Conclusions

The PRISMA organization



Departing situation

- Emergent decision of using UML
- Use case specification started once a first prototype was already running
 - Not intended to drive system development but:
 - Systematic validation, communication, maintenance, prestige (“UML-label”)
- Developers were the analysts...
 - Had less time than required to write use cases
 - Did not perceive use cases as important as implementation

Decisions made

- Prioritise use cases (we had almost 800!)
 - Units = functional areas
- Write a prescriptive & precise use case template
 - Emphasis on data
- Define a methodology to develop the use cases
- Design tool support for the methodology
 - Visure RMT (diagrams) + Lotus Notes (text)
- Design a procedure for change management
 - Technical and organizational

Lessons on use case design criteria

- Do not get rid of the existing implementation
 - Take into account the general structure of the solution
 - Define packages corresponding to main system areas
 - Use navigational maps and forms
- Give preference to robustness over accuracy for dealing with small variations of business processes
 - Use abstract actors
 - Unify essentially similar processes

Lessons on project management

- Prioritize use cases if resources are insufficient
 - Explicit task of prioritization
 - Define actionable criteria for prioritization (strategic relevance, stability were used in our case)
 - Prioritize logical units of use cases
- Do not start to write the use case specification until a template has been defined
 - Uniform style
 - Productivity increases
 - Facilitates writing a checklist to verify use case properties

Lessons on project management (cont.)

- Introduce use cases smoothly in the organization
 - Involve developers by encouragement instead of obligation
 - Provide handy templates, methods, procedures...
 - Do not force to learn new tools further than necessary
- Monitor continuously use case models to reduce risks
 - Abstract from the user interface details
 - Ensure appropriate scope (business-oriented)
 - Continuous update as system evolves

Lessons on use case technical aspects

- Introduce mechanisms to define the structure of system data and a glossary for domain-specific terms
 - Use case specification self-contained
 - Coherence among different use cases
 - Use cases text shorter
 - Supports validation with existing GUI forms
 - Supports maintenance
- Do not abuse of use case relationships
 - Make the specification difficult to understand
 - Workflows related to change management get complicated
 - Raise some technical points (linkage, preconditions...)

Conclusions

- The experience can be considered successful
- A complete use case model was built fulfilling the goals
- PRISMA developers changed their mind!

Thank you!



Group of Software Engineering for Information Systems

UNIVERSITAT POLITÈCNICA DE CATALUNYA