Roles and role conflicts in reuse projects

Even-André Karlsson

Q-Labs
Ideon
S-223 70 Lund
Sweden
Email: eak@q-labs.se

Abstract

Reuse is seen as one of the main sources of productivity increase in software development within the next decade. Up till now there has been mainly a focus on either organizational or technical aspects of reuse. We argue that to achieve reuse we have to organize and plan both development for and with reuse within a project. In this article we identify different roles in connection to reuse, and the role conflicts that leads to many of the reuse inhibitors which block the full potential of reuse. This work is done within the context of the ESPRIT project REBOOT. The REBOOT projects aims at producing methodologies and tools supporting reuse, but is also giving considerable attention to the equally important implementation of reuse in industrial environments.

Keywords: Reuse projects, roles, conflicts

Workshop Goals: Learning; networking; discuss ideas

Working Groups: reuse project management, reuse organization

1 Introduction

Reuse has attracted considerable attention lately as one of the main potential for productivity increase in software development. Up till now the focus on reuse has mainly been on either technical or organizational aspects. Techniques for e.g. domain analysis, classification and development for and with reuse have been extensively covered, as well as how an organization shall be structured to achieve reuse, including incentives and education. All these aspects are important, but we think that the even more important role of the project manager, and how he has to organize his project to achieve reuse, has been neglected. In this paper we will discuss his role, and how he can analyse his project, its stakeholders and their conflict to achieve reuse.

We can distinguish two main forms of reuse activity:

- Development for reuse which is concerned with making components which can be reused by others. Here we must identify the scope of the components, i.e. how general is it optimal to make it. This is mainly done by analysing the needs and benefits for po- tential customers and reusers versus the cost of the additional generalizations.
- Development with reuse which is concerned with finding, evaluating and adapting the general components. Here we have to make a cost benefit analysis on whether it is cost effective to reuse a component instead of developing it from scratch. We also have the possibility here to negotiate with our customer based on the available components.

Both forms of reuse activity will exist in most mature reuse organizations, but we can envision organizations only specializing in one.

A mature reuse organization consists of different projects and a set of reuse assets. The reuse assets are seen as a long time investment reflecting the company strategic directions. The organization can, based on strategic choices, decide in which product lines to invest (development for reuse) and for which areas only development with reuse is interesting. Within this framework it is important that the project managers know how to conduct the different kinds of projects, and what are the pitfalls. The project manager has a focal role, as he has to reconcile all the conflicting interests which are present in a reuse project.

The project manager's role is becoming more complex in a mature reuse organization, as there will be many more stakeholders to relate to than in an ordinary development organization. In development for reuse there will be more customers, and in development with reuse there is uncertainty on how much can be reused.

2 Roles in the reuse project

In this section we will analyse the different roles which are stakeholders in reuse projects. The roles involved in reuse projects are suitably analysed using four views, management, development, customer and support. For each view we also try to characterize what is different for a reuse project, divided into development for and with reuse.

For a reuse project it is important to identify which roles are present, and to agree which persons represent the different roles. There is no guarantee that this list is complete, and for any persons not allocated a role, it is important to analyse what interests he has in the project.

It is helpful to identify which role a person represents when one is engaged in communication. By this knowledge many common communication misunderstandings can be avoided.

2.1 Customer view

The customer view represents the projects interface to its customers. The different customer roles are explained in the sequel.

Development for reuse In development for reuse there are several customers (some real and some assumed) all which requirements we shall incorporate in the development. In the customer view we must also take into consideration the ones which are to adapt the reusable components. Thus the different customer roles are:

- Identified customers are those which are waiting for the reusable components when they are under development. They are able to express their requirements and validate that they are taken into account. Note that there is one customer role for each potential customer of the system as they represent different requirements.
- Potential assumed customers are any future reusers of the components. Their requirements are much more difficult to guess, and are thus more dif- ficult to take into account.
- Identified reusers are the application developers who shall specialize the components. We distinguish between customers which are mainly interested in the functionality of the final specialized components and those who are to make the specialization.
- Potential assumed reusers are any future reusers of the components.

Development with reuse In development with reuse there is only one customer, but here we have the possibility to negotiate the requirements based on the available components.

Application customer is the one interested in a given application for which we want to reuse an existing application. If there is not a complete match between the required functionality and the one offered by the reusable component, we can either adapt the component or negotiate the requirements with the customer. Note that the earlier in the requirements capturing process we find potential reusable components, the easier it will be to adapt the customer's requirements.

2.2 Development view

The development view represents the transformation of the customers' requirements into a final product, including support and maintenance.

In each of development for and with reuse there is only one role:

- Reuse developers are to develop generalized solutions which incorporate all the different requirements. The solution must provide a positive cost benefit ratio for all potential reusers and in total be larger than the initial development. For each additional requirement we must analyse how much benefit it provides to the customer versus how much it will decrease the benefit of the other reusers. We must also here take into consideration the performance as well as complexity of the components.

- Application developers with reuse are to develop specialized applications, and to their disposal they have a set of reusable components. They have the choice of either implementing the functionality from scratch or attempting to reuse and if necessary adapt existing components.

Support and maintenance is provided by the maintainer. Maintainers are responsible for correcting errors, providing functional enhancements and supporting the reusable component. Maintenance is the grease between producers and consumers being a guarantee for the quality of the component.

2.3 Management view

The management view consists of the project managers and those she interacts with outside the project.

- Project manager is the one responsible for the project. She has a finite set of resources allocated, primarily time and manpower, and shall achieve a set of goals. She has to plan the project and allocate the resources so that she can achieve as many as possible of the goals.
- Resource allocator is the one allocating resources to the project. Initially the project is given some resources, but this allocation can be changed during the course of the project. These changes can be due to a change in goals or that the initial resources where to plentiful or to scarce. Development with reuse is always a risk as it is impossible to fully predict the amount of work needed to adapt a component for reuse. It is important that this risk is well understood, and that sensible judgement is conducted, but there is no guarantee that there will never be misjudgements resulting in overruns.
- Project evaluator is responsible for assessing the project both as it is running and when it is over. This can be both with regards to project planing (process) and produced results (product).

2.4 Support view

The support view consists of line organization roles which can be useful for a reuse project. For each role we have divided the discussion into development for and with aspects.

Reuse development methodology expert Development for reuse experts are proficient in general techniques for representing variability at different levels of abstraction, from requirements to code. This also includes how to document the intended reusability.

Development with reuse experts support general techniques for adapting components into applications. This includes adapting the component so that as little as possible need to be changed regarding more concrete abstraction levels (e.g. design, code and test if we reuse analysis) and documentation.

Cost benefit analyzer In development for reuse we must weigh the potential benefits of adding additional requirements against the costs. The benefits are the hardest to estimate as they depend on the future reuse of the components, and for future assumed customers this is difficult to estimate.

In development with reuse we must weigh the cost of adapting the components or its environment or developing the functionality from scratch if there is not an exact match. This is a difficult decision where the main problem is to evaluate the amount of change needed. In particular if we reuse large components early in the life cycle a wrong decision can have catastrophic impacts.

Application domain expert In development for reuse an application domain expert is someone knowing the application domain, i.e. the customers world. She might be a product line manager, and will also be able to make predictions or strategic choices for what functionality which shall be represented in the assets.

In development with reuse there is not that much need for a domain expert, as we are now to satisfy the specific requirements of one particular customer.

Asset expert In development for reuse an asset expert knows existing components within the domain, and specific guidelines for how they are to be developed. He will also be able to ensure that the functionality of the new reusable components will fit into the existing set of components. He can also have a quality assurance role for the new components.

In development with reuse an asset expert knows the existing components within the domain, and are able to guide the reuser to the right component based on their specific requirements.

Library expert In development for reuse a library expert shall ensure that the new components are inserted correctly into the library so that they can be retrieved by potential reusers. She will also be responsible for introducing new terms if that is found necessary.

In development with reuse the library expert will assist the reusers in retrieving a set of candidate components which can be evaluated, and possibly reused.

3 Role conflicts

In this section we identify some of the conflicts which can occur between different roles in a reuse project. These role conflicts will result in many of the well known reuse inhibitors which block the full benefits of reuse. For each of the role conflicts we will try to indicate possible counter measures, thus reducing or removing the reuse inhibitors.

A role conflict will occur when there is a difference in interests and there are not enough resources to satisfy both sides. When this situation occurs it is important to be able to analyse the possible solutions from an objective standpoint. To do this it is important to determine which roles the different persons involved in the conflict represents, and both their long and short term cost and benefits of different solutions. It is often the case that the same person will have more than one role. It is then often difficult to get an objective view of the conflict, and make the most rational solution. In particular if some of the roles are more important than others.

We have divided the discussion of the role conflicts into development for and with reuse. Most of the conflicts appear within and between the different roles in the customer and development views. The other roles are mostly used to represent the weak part in these conflicts, and as an arbiter.

3.1 Development for reuse

The main problem in development for reuse is to find the right level of generalization; a too general component will be too complex and difficult to understand and specialize, whereas a too specific components will not fit many reusers. As we collect the requirements from different potential customers we must always keep this trade-off in mind, as every role will try to maximize its benefit from the component.

It is not a trivial to find the right level of generalization which satisfies all parties with respect to functionality, performance, complexity, maintainability and adaptability. There will always be conflicting interests when several different parties are involved.

The conflicts do usually surface when the resources gets short, i.e. we are already late, up till them people have been shuffling the problems ahead, hoping that they will disappear. At this point it is difficult to find any satisfactory solution for all parties. It is therefore recommendable that we analyse the stakeholders involved at an early stage to get the conflicts into the open.

Customer — customer conflicts Different customers will to some degree have differing requirements - otherwise they would be the same customer. These differing requirements will lead to conflicts when it comes to what the software shall provide. The aim of development for reuse is to identify common requirements which are invariant for many customers, and design components which reflect these and allow each customer to specialize the components for his particular needs. All customers will like the reusable components to be as close to their total requirements as possible, thus minimizing their own effort in specializing the component.

These conflicts are particularly dangerous when the strength relationship between the different customers is uneven. This is for instance the case between identified and potentially assumed customers, and in a development for reuse within an application project where the application customer is in command. In these cases it is important that we find someone which can represent the weaker interest, e.g. an application domain expert.

Designer — designer conflicts When we have brought the conflicts in the requirements into the open it is important that we identify the benefits of the different requirements so that the designers can take the most important into consideration. The designers will thus have many of the same internal conflicts as the customers, i.e. the designers which have specific interests in reusing the components will ensure that it fits them as much as possible. In many cases the different roles of the reusers will be represented by only one developer, and if he has his own interests as well the other reusers will suffer. This is particularly common during development for reuse in application projects.

Customers — designers conflicts The customers (and potential reusers) would like the system to be as fine grained when it comes to representing the variability to minimize specialization effort, whereas the designers wants to have a coarse grained representation which is easier to develop and maintain. This can be exemplified with a class hierarchy, the customers wanting a fine grained and deep hierarchy, whereas the developers will strive for a coarse grained and shallow. But even for the reusers there will be an advantage of a more coarse representation as it will be easier to understand. This conflict is continued between the reusers and maintainers of the components.

There is also a second source of conflict concerning variability which relates to combination of different functionality, i.e. someone wants to combine two pieces of functionality which were not designed together. This problem is similar to feature interaction in telecom systems and multiple inheritance problems in object-oriented programming. Thus for every pair of differing requirement we should perform an analysis to see if they can be needed together, and if that is possible in our design.

3.2 Development with reuse

The main problem in development with reuse is to know when to reuse. Is the effort to search for, evaluate, understand and possibly adapt a reusable component less than the effort used to develop it from scratch? Note that the process of reusing a component is divided into clearly separated steps, where each step requires effort. The savings for the reused components, must cover all these steps (also for the instances where we stopped the process discarding all candidate components) as well as the investment in developing the reusable components.

Customer and developers Application customers and reusers will have a potential conflict if the reuser has found a component which fits some but not all of the customers requirements. The reuser will save considerably on reusing the component, but the customer will not get all his requirements fulfilled. It is then a negotiation process between the customer and the reuser to change the requirements.

Reusers and maintainers Reusers are responsible for extracting and specializing reusable components, whereas maintainers are responsible for correcting bugs and when necessary enhancing the component. There is a gray zone between what is enhancement which should be done to the general component, and what is specialization. If the reuser can convince the maintainer to enhance the component so that he can reuse it as is the reuser saves both the development and maintenance of the specialization. It might even be cost effective as the maintainer will know the component better than the reuser.

Reuse developers and reusers Between developers of reusable components and reusers there need to be, as with any other piece of software, a relation of trust. There is always a tendency to regard any piece of code which one has not written oneself (with one's owns standards) as a risk — the not invented here syndrome. To increase the trust reusable components must have a specified quality level, and to ease understandability it should follow development and documentation standards. If possible potential reusers should be able to get in personal contact with the developers or maintainers of larger components, preferably before they commit to reuse it.