Viewing Project Collaborators Who Work on Interrelated Requirements

Irwin Kwan, Sabrina Marczak and Daniela Damian
Software Engineering Global Interaction Lab (SEGAL)
University of Victoria
3800 Finnerty Rd., Victoria, B.C., Canada
{irwink, smarczak, danielad}@cs.uvic.ca

Abstract

Project collaborators in a software development project need to stay aware not only of changes to requirements and other artifacts, but also of each other's current work. The set of team members working on a requirement is dynamic, and team members who were not assigned to the requirement in the plan may be involved. If this requirement changes, those team members who are dependent on that requirement must be notified quickly before they do outdated work. However, project plans often do not provide an easy method of listing all of the emergent team members who should be notified of changes to a requirement. We propose a requirements-dependency diagram that displays interdependent requirements and team members who are assigned to these interdependent requirements. The visualization highlights prominent collaborators, lists each collaborator and each requirement only once, marks emergent collaborators, and is simple and clutter-free. By viewing this diagram, collaborators will know who to contact to notify others of changes to requirements, and can contact experts working on interrelated requirements.

1. Introduction

A software development project requires good communication among its team members to ensure its success. One communication message that is important to propagate in a timely and efficient manner is requirement-change information. If a requirement changes, each team member who is working on that requirement, and dependent requirements, should be informed [1]. If a team member uninformed about the progress of a requirement continues to do work based on outdated information, the team may be required to rework not only requirements, but also documentation, and source code. It has been shown that changing the requirements in the middle of a release cycle can cause delays in requirements [4].

During software development, more developers than initially planned can be come involved in the requirements [1, 3, 2]. The dynamic nature of the team means that it is difficult to maintain contact with each team member. Another issue in software development is that project collaborators may not be aware of which requirements are affected by their work, or which requirements have an effect on their work. This is especially the case if team members are remote, and emergent [2].

In order to help maintain awareness of who else is working on requirements that are related to a team member, we propose a visualization that lists a set of interrelated requirements, along with the collaborators who work on these requirements.

2. A Visualization of Interdependent Requirements

In order to better present who is working on which requirement, and which requirements are interrelated, we designed a visualization that displays requirements as circles, and project collaborators as names on the diagram (Figure 1). The requirements encircle the team members. This representation provides information about the set of requirements and the team members assigned to work on these requirements. It also shows the team members who may work on one or more of the requirements. A focus person who is selected has his name highlighted in bold, and the requirements he is working on are highlighted as well. In Figure 1, we view the diagram from Robert’s perspective, and see that he is working on Requirement R8. In this diagram, R8, R4, and R6 are interrelated. George works on both R6 and R8. Ted works on R4 and R6, and Perry works on all three requirements.

We designed the diagram with the following properties.

Inter-dependent requirements are displayed together. We display every dependent requirement with overlapping
shapes to indicate inter-dependencies, reducing the need to browse multiple pages for dependency information regarding the requirements. The current requirement of interest is bolded for better visual recognition.

Prominent team members are in the centre. Team members who are involved with every requirement in the set appear in the centre. In our experience, the team members who work on every requirement are most-often experienced individuals who are important to the team. In the event that a team member cannot contact everyone who appears in the diagram, he can contact the experienced team members who can take additional action.

Each collaborator and each requirement appears only once. Other visualizations of dependent requirements may require that the requirements or the actors be listed multiple times. This representation ensures that each name and each requirement appears only once, reducing the cognitive effort required to view the diagram.

Emergent collaborators are marked. If a team member who was not originally assigned to work on the requirements becomes involved in the requirement at a later time, we can identify that team member as someone whom the team should be aware of. We mark that person’s name with a preceding exclamation mark. In Figure 1, Lambert is an emergent collaborator.

The diagram is simple and clutter-free. The lack of edges in the diagram makes the diagram easy to read. However, hypergraph representations become increasingly difficult to lay out as the number of interrelated requirements increases. It is an area of future research to see how we can best display complex interrelated requirements.

2.1. Application to a Development Environment

In a digitized version of this diagram, we can link each user to additional information about each requirement or each person. We can implement quick-access buttons to allow a collaborator to contact each person in the diagram, or to explore additional requirements in the project.

There are multiple methods to gather information regarding requirements in order to generate this diagram. We can generate the diagram using a tool that mines communication information. A tool that tracks assignments and communications in a problem tracking system would be able to provide the information necessary to update this diagram.

3. Conclusion

A collaborator must know who is affected if he makes a change to a requirement. We suggest a visualization of interdependent requirements that allow a user to identify collaborators who work on interrelated requirements.

The visualization is meant to 1) highlight prominent collaborators, 2) list each collaborator and each requirement only once, 3) mark emergent collaborators, and 4) be simple and clutter-free. These traits allow the diagram to be used by developers to keep in contact with collaborators and seek experts, or by managers to ensure announcements and notifications reach the entire group who is working on affected requirements.

References