The Groupthink specification exercise

Michael D. Ernst
MIT Computer Science & Artificial Intelligence Lab
mernst@csail.mit.edu

John Chapin
Vanu, Inc.
jchapin@vanu.com

Abstract

Teaching students to read and write specifications is difficult. It is even more difficult to motivate specifications — to convince students of the value of specifications and make students eager to use them. This paper describes the Groupthink specification exercise. Groupthink is a fun group activity, in the style of a game show, that teaches students about specifications (the difficulty of writing them, techniques for getting them right, and criteria for evaluating them), teamwork, and communication. Specifications are not used as an end in themselves, but are motivated to students as a means to solving realistic problems that involve understanding system behavior. Students enjoy the activity, and it improves their ability to read and write specifications. The two-hour, low-prep activity is self-contained, scales from classes of ten to hundreds of students, and is freely available to other instructors.

Categories & Subject Descriptors: K.3.2: Computer and Information Science Education: Computer science education; D.2.1: Software Engineering: Requirements/Specifications; C.0: Computer Systems Organization: Systems specification methodology
General Terms: Management, Documentation

1. Introduction

Specification (along with related activities such as testing and verification) is critical to the success of any real software system. However, many students view specification as a dry, tedious, and impractical topic.

One problem is that many undergraduate programming classes fail to integrate specification into the curriculum in a realistic way. Grading specifications takes them out of the context in which they will be used, and students gain no appreciation for their utility. Additionally, typical class assignments are simple, so techniques that are crucial for more complex software may not be cost-effective. Students learn the (incorrect!) lesson that specification is pointless.

We have designed a two-hour group activity that teaches students about specification — conveying its utility, illustrating how to do it (and how not to do it), and emphasizing the importance of teamwork and communication during the process — through hands-on experience with a realistic problem. We have run the activity 7 times, involving over 600 sophomores from the MIT School of Engineering, as part of the January inter-session UPOP (Undergraduate Practice Opportunities Program) engineering "boot camp" [1].

ICSE'05, May 15–21, 2005, St. Louis, Missouri, USA.

Our three major goals in designing the activity were as follows. First, we wished to pose students a well-motivated, practical, concrete, realistic problem. The problem needed to be small enough to look easy at first glance, but hard enough that undisciplined approaches were unlikely to be successful, so that the students appreciate the advantages of using a specification. Second, we wanted to run an interactive and lively event. Many people learn best by doing (e.g., active learning), and hands-on activities are more likely to be fun and to inspire their participants. Iteration gives students an opportunity to learn from and to correct their mistakes, gaining a feeling of accomplishment. Interaction with other people exposes students to multiple technical and organizational approaches to a problem. Third, we aimed to appeal to participants with many different backgrounds. The activity should equally engage computer science students and computer-phobes, and those with years of job experience or none at all. This implies that the activity must teach different skills and concepts to different people, and it should encourage participation by all students. For example, we wished to force "techies" to talk to "non-techies", which teaches better communication skills to both groups.

In sum, the Groupthink specification exercise teaches students how difficult it is to write an effective specification (determining what needs to be specified, making the choices, and capturing those choices) and gives them practice in doing so, in a fun environment that is conducive to learning.

2. The activity

The activity starts with a very brief lecture on specifications. Students then form teams of 7–10 people and are given a partial specification of a system, plus a set of requirements for its behavior. The specification is incomplete and inconsistent in somewhat subtle ways. Each team is asked to develop a better and complete specification. There is no right or wrong specification as long as the requirements are met. The goal is for each team to agree upon the specification and to understand what they have agreed upon.

The system we used is a desktop telephone with integrated answering machine. Other systems would also work well. The system should be relatively familiar to students to minimize confusion, and should have a moderate level of feature interactions to make the specification rich but tractable.

To evaluate their specifications, students play the Groupthink game. Teams are asked a series of questions about the system behavior that focus on the feature interactions. A sample question is:

> The user is connected to an outside party. The outside party hangs up. What state is the phoneline in?
> A. Lineactive (the user hears dialtone)
> B. Lineidle (the user does not hear dialtone)

Each team member individually answers the questions. There is no right answer. Instead, the challenge is for all team members to give the same answer, without communicating with one another, based on the specification they have developed. It is important to emphasize that differences in answers are a failure of the group to develop a specification that everyone understands, not a failure of any individual.

Each group receives one point for each answer with a plurality of votes. For instance, if 3 team members answer "A", 4 answer "B", and 2 answer "C", the team receives 4 points. A bonus of 10 points is earned if all answers agree. Displaying a running tally of scores promotes excitement and involvement in the game. Winning or losing the game depends primarily on the number of bonuses earned.

After a round of questions, there is a group discussion that conveys the core of the material the students will learn (see Section 3). Then, though they are not told this beforehand, the students are given more time to improve their specification before a second round of the game. Students use their mistakes from the first round to help them improve. This learning experience is key: students get to make mistakes, learn from them, and then do a better job, which solidifies the learning and leaves them with a positive feeling.

After the game, the activity concludes with a lecture that reinforces the points made by the introductory lecture, the discussion in the middle, and the exercise.

3. Lessons learned

After the first round of the Groupthink game, there is a group discussion in which students explain what strategies were and were not successful. We believe that students hearing from one another in their own words is more effective than a lecture. An instructor moderates the discussion, following up on certain points and drawing connections.

Student remarks tend to fall into two categories: technical approaches and group organization. Some common technical approaches are:

- Rank all actions: give some priority over others.
- Draw state diagrams with transitions among states.
- Draw state diagrams showing the state of all components (switches, phoneline, etc.).
- Explore a set of "use cases", indicating what happens in each circumstance.
- Decompose behaviors into normal and exceptional cases, and treat each separately.
- Focus the specification around the communication protocol among system components.
- Create a rule-based system with a set of (prioritized) rules.
- Divide and conquer: delegate different parts of the specification to different sub-teams.

Common approaches to group organization include:

- Choose a leader to direct and focus the group's activity.
- Don't argue. To avoid wasting time, poll everyone, make a decision, and then move on.
- Choose simple behavior that is easier for everyone to understand; avoid bells, whistles, and complicated features that are not explicitly noted in the requirements.
- Listen to everyone — don't shut out some participants. Often quiet group members had very good points to make, and those with more forceful personalities needed to listen carefully.
- Ensure that everyone understands — the group is only as strong as its weakest member. Group members comfortable with the technical terms used in the assignment needed to help the others. Students come to realize that although the exercise is couched in telecommunications and computing terminology, it does not rely on skills specific to those disciplines.
- Use a master copy so everyone can focus on it. When this is on the whiteboard, more people tend to participate then when it is on paper, and it is more useful than each person writing up a separate copy.
- Spend a few minutes in which everyone reads the handouts before diving into a solution, in order to guarantee understanding of the problem.

4. Assessment

The Groupthink specification exercise has been a success. Students eagerly and intently attack the problems during the activity; they loudly cheer and groan as their team's fortunes change during the game; and they confirm their enjoyment in a post-activity survey. Students reported learning a wide range of ideas and skills during the activity. The five most common ones that students reported were communication, simplicity, teamwork, clarity, and the importance of having a written-down specification. Student performance improved during the game. We compared scores during the first round of play (before students had observed their performance and had a group discussion about various technical and organizational approaches) and the second round. The improvement ranged from 4% to 131%, with an average of 66%. Student self-assessment indicated that, compared to a control group, the UPOP students had greater improvement in their teamwork abilities [1]. (Before UPOP, 18% of students reported never having worked as part of a team either in class or on a job; 36% reported one such experience; and 46% reported 2 or more such experiences.)

We attribute the success to the design of the activity, which paid careful attention to the goals noted in Section 1. The problem was realistic enough to motivate students, hard enough to require use of specifications, and easy enough to give students a sense of accomplishment. Students got a chance to fail in an initial attempt, then a chance to succeed after reflecting on their experience and learning about a wide variety of other approaches taken by other teams. Achieving high scores required full participation from every team member; this motivated students to communicate and explain their ideas in a simple and clear manner. Students with different backgrounds learned different lessons, whether technical, communication, or organizational.

The Groupthink specification exercise is available on request from `mernst@csail.mit.edu`. It includes lecture slides, a handout describing the answering machine requirements and partial specifications, the game show questions, a scoring spreadsheet, and a detailed instructor's manual. Instructors outside MIT have successfully used the materials. Because the activity is self-contained, it can easily augment an existing curriculum, for instance to motivate specifications or to replace dry lectures. Alternately, by incorporating more instruction about specifications, the problems presented in the game could be made more challenging.

5. References

[1] C. Leiserson, B. Masi, C. Resto, and D. K. P. Yue. Development of engineering professional abilities in a co-curricular program for engineering sophomores. In *ASEE Annual Conference*, June 2004.