

**Educators' and Trainers' Symposium**  
**Active Learning Exercise**  
**Students' Cooperation in Teamwork:**  
**Binding the Individual and the Team Interests**

Orit Hazzan  
Department of Education in Technology and Science  
Technion – Israel Institute of Technology  
Haifa, Israel  
oritha@tx.technion.ac.il

Yael Dubinsky  
IBM Research - Haifa Lab  
Mount Carmel  
Haifa 31905, Israel  
dubinsky@il.ibm.com

### Abstract

This Active Learning Exercise aims at increasing the participants' awareness to the importance of cooperation in software teams as well as at guiding instructors of project-based courses in the evaluation process of students' projects. The Active Learning Exercise is based on individual activities, teamwork activities, discussions and reflections.

**Categories and Subject Descriptors** D.2.9 [Software Engineering]: Management – *Programming teams*

**General Terms** Management, Measurement, Performance, Design, Economics, Human Factors, Legal Aspects.

**Keywords** human aspects of software engineering, teamwork, bonus, reward, grading.

### 1. Introduction

The Active Learning Exercise aims at increasing the participants' awareness to the importance of cooperation in software teams as well as at guiding instructors of project-based courses in the evaluation process of students' projects. The activity consists of three main sections. First, connections between bonus allocation and cooperation are investigated. Second, cooperation in software development processes is examined through the Game Theory framework of the Prisoner Dilemma. Third, based on the understandings gained in the first two sections, a grading policy is constructed, addressing the individual interests, the team interests and the unavoidable need to cooperate in the development of software projects. The Active Learning Exercise is based on individual activities, teamwork activities, discussions and reflections.

### 2. The Active Learning Exercise

**2.1. Bonus Allocation** (Hazzan, 2003; Tomayko and Hazzan, 2004)

This part of the Exercise illustrates how students' cooperation can be increased by introducing them to win-win situa-

tions. It consists of three stages, each one is followed by a reflective session.

#### **Step 1: Individual work**

**Task:** Assume that you are a member of a software development team. Your team is told that if the project it is working on is successfully completed on time, the team will receive a bonus. Five options for bonus allocation are outlined below (See Table 1). Please explain how each option might influence team cooperation, and select the option you prefer.

**Table 1. The task**

	<b>Personal Bonus (% of the total bonus)</b>	<b>Team Bonus (% of the total bonus)</b>
a	100	0
b	80	20
c	50	50
d	20	80
e	0	100

#### **Step 2: Team work**

Each team decides on one option that the team members, as a team, prefer.

#### **Step 3: Individual work – reaction to two situations**

A. Your supervisor tells each of the team members, **separately**, that if he or she performs better than the other team members, he or she will be promoted. The team members do not know that each of them is told the same.

- In your opinion, how will this effect team cooperation?
- If you were one of the team members, how would you suggest sharing the bonus now?
- How would you behave in such a situation?

B. Now, your supervisor tells each of the team members, **separately**, that his or her contribution to the teamwork is a major factor contributing toward his or her promotion. The team members do not know that each of them is told the same.

- In your opinion, how will this effect team cooperation?
- If you were one of the team members, how would you suggest sharing the bonus now?
- How would you behave in such a situation?

**Activity explanation**

The task is composed of three steps. Step 1 focuses on the participants' preferences when a neutral situation is described. Step 2 examines how they face possible conflicts between their own preferences and the preferences of the other team members. Before proceeding with Step 2, the participants' written responses to Step 1 are collected in order to ensure that the answers to Step 1 were not changed later on.

Step 3 presented the participants with two cases. The first addresses a situation in which the participants have a personal incentive; the second describes a situation in which there is an incentive to contribute to the teamwork. In both cases, personal promotion is conditional.

Following the completion of all three stages, a discussion takes place. The participants share their feelings, conflicts, and rationales for choosing a particular option of bonus allocation in each scenario. One of the main lessons highlighted during this discussion is that cooperation is vital in software development processes. The discussion is continued with an analysis of the topic of reward allocation by theories taken from Game Theory, mainly the Prisoner Dilemma.

**2.2. The Prisoner Dilemma: The Case of Cooperation in Software Teams**

See Tomayko and Hazzan (2004) and Hazzan and Dubinsky (2005) for this analysis.

**2.3. Grading Policy for Student Evaluation (Hazzan and Dubinsky, 2003, 2008)**

The two previous parts of the activity inspired the message that in software development environments the individual interests are bound to the team interests and one cannot achieve his or her targets without taking into the consideration the team interests as well. Further, one theme which enables this binding is cooperation.

At this stage we illustrate how this understanding, which refers to the linked nature of the individual and the team interests in software team, can be reflected in a grading policy constructed for the evaluation of students' software projects.

In general, it is accepted that when a university courses instructor wishes that his or her students follow specific principles that he or she deems important, these principles must somehow be incorporated into the evaluation policy of the course. This is particularly true when a software development method is used in a project-based course. It is reasonable to assume that students naturally devote more effort to what is valued (and graded).

Among different options, we present here a grading policy which has this property, as is described in what follows.

According to this grading scheme the grade is composed of an individual component (35%) and a team component (65%), which was identical for all members of the team (see Table 2). Naturally, such an evaluation scheme conveys the message that both teamwork and individual contribution count. In practice, students are encouraged to contribute to the teamwork on the one hand, and on the other hand, this evaluation scheme affords those wishing to excel, the opportunity to improve their grade through the personal component of the grade.

**Table 2. Example of a cooperation-oriented grading policy**

Team Component (65%)	Individual Component (35%)
<b>60% -</b> Answer the customer stories and meeting the schedule according to the team time estimations: ✓ <b>(10%)</b> for iteration 1 ✓ <b>(25%)</b> for iteration 2 ✓ <b>(25%)</b> for iteration 3  <b>25% -</b> Project documentation  <b>15% -</b> Team evaluation by the academic coach	<b>40% -</b> ✓ Weekly reflection ✓ Student-supervisor pair programming experience ✓ Test-Driven-Development exercise ✓ Weekly presence  <b>40% -</b> Performance of a personal role: ✓ Actual implementation ✓ Further development and enhancement  <b>20% -</b> Individual evaluation by the coach

Clearly, instructors can adjust the ratio between the individual and the team components and the specific ingredients of each component according to their teaching goals.

**3. Conclusion**

This activity suggests utilizing the conflicts and dilemmas rooted in bonus allocation task and the Prisoner Dilemma analysis, as a means for the construction of an evaluation scheme for students' software projects. A similar activity is conducted by us also with software development teams in the software industry.

**4. References**

Hazzan, O. (2003). Computer Science students' conception of the relationship between reward (grade) and cooperation, *Proceedings of the Eighth Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2003)*, Thessaloniki, Greece, pp. 178-182.

Hazzan, O. and Dubinsky, Y. (2003). Teaching a Software Development Methodology: The Case of Extreme Programming, *The proceedings of the 16<sup>th</sup> International Conference on Software Engineering Education and Training*, Madrid, Spain, pp. 176-184.

Hazzan, O. and Dubinsky, Y. (2005). Social Perspective of Software Development Methods: The Case of the Prisoner Dilemma and Extreme Programming, *Proceedings of the Sixth International Conference on Extreme Programming and Agile Processes in Software Engineering*, Sheffield University, UK, pp. 74-81.

Hazzan, O. and Dubinsky, Y. (2008). *Agile Software Engineering*, Springer.

Tomayko, J. and Hazzan, O. (2004). *Human Aspects of Software Engineering*, Charles River Media.