

# Teaching Software Development Process Using the Team Collaboration Platform

YEVGENIA KOZMINA

Vocational Training Technologies Department

Saint-Petersburg State University of Information Technologies, Mechanics and Optics

49, Kronverksky St., Saint Petersburg

RUSSIA

[yekozmina@gmail.com](mailto:yekozmina@gmail.com)

*Abstract:* The Software Engineering program curriculum in Russia lacks both real practice in the area of software development and teaching of software development process in general. This paper reports on a solution which can be used to enhance interest and retention in the existing Software Engineering curriculum through the use of game-based models of real development examples.

*Key-Words:* Collaboration, Software development process, Tools.

## 1 Introduction

Software development courses in Russia often don't include such an important aspect as the software development process. As a result graduate students have knowledge of basic software engineering principles, but can't understand basic and auxiliary software development business processes and their roles in these business processes. Often students have only superficial understanding of the software development process without reference to concrete activities, therefore in the beginning of their career they can't self-identify themselves.

It is a common problem when such aspects as: the software development life cycle, methodologies, process organizational structure, professional software, utilities, platforms, software developments process actors and their professional and social interactions — are not considered in the educational process.

However such social software development factors as: cognitive characteristics [1], [2], [3], group dynamics and team composition [4], [5], coordination in software development [6] — are considered to be the most vulnerable.

As it is hard to teach students social interaction using just a standard set of lectures and practical lessons, it is necessary to allow students to organize in communities or teams, provide them with the necessary tools, a set of methods and organize consultations. Thus students can get their own experience, understand common mistakes and

develop the best solution in organizing the software development process for them.

## 2 THE TEAM COLLABORATION PLATFORM CONCEPT

The team collaboration platform concept is based on a coordination theory. The coordination theory is an interdisciplinary theory and is currently at the phase of formation. In the coordination theory there are such entities as frameworks, hypotheses based on analogies with adjacent and similar disciplines, psychological, social and other constructions.

The term 'coordination' is defined in different ways. In an article 'Interdisciplinary Study of Coordination' [7] authors give the following definitions:

- the operation of complex systems made up of components;
- information processing within a system of communicating entities with distinct information states;
- the emergent behavior of collections of individuals whose actions are based on complex decision processes;
- the joint efforts of independent communicating actors towards mutually defined goals.

In this article the term 'coordination' is defined by context of the theory of multi-level hierarchical systems [8]. Authors propose to use the theory in the case when there are elements of three organization theories: classical, behavioral and system-oriented. As in the system of the software development

process organization where there are features of all three theories it is possible to make an assumption that the theory of multi-level hierarchical systems is applicable to software development. The theory's authors do not give a strict definition of the term 'multi-level hierarchical system', however list its characteristics: vertical subordination, right of intervention and interdependence of actions. With a careful analysis software development system can be attributed to these systems.

The authors of this theory understand coordination as a combination of such aspects as self-organization and management. According to these aspects in multi-level hierarchical systems three levels can be defined: the level of choice, the level of education or adaptation and the level of self-organization.

For software development these layers can be represented this way.

- The level of choice is represented by task or problem formulation. The task can be represented by:
  - a Vision&Scope document or/and a Business Requirements document,
  - a specification or a project,
  - a set of requirements,
  - an agreement and its annex.
- The level of adaptation can be represented this way:
  - software development methodology,
  - a set of tools used in a project,
  - a set of requirements, scenarios and similar entities if they are not represented in the level of choice.
- The level of self-organization can be represented this way:
  - meetings,
  - coding, testing, analysis and similar activities,
  - consultations,
  - planning.

Therefore all three levels should be represented in developed the team collaboration platform. Also it is necessary to establish connections between entities of all tree levels.

### 3 THE TEAM COLLABORATION PLATFORM COMPONENTS

The team collaboration platform allows organizing students' software development activities.

A general scheme is shown in Figure 1.

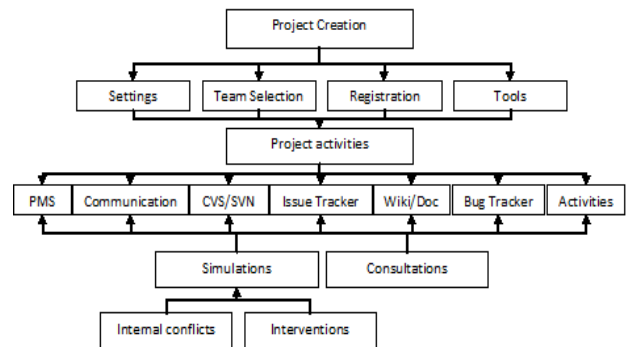


Fig.1: A general scheme of team collaboration platform components

Below you can see brief descriptions of each team collaboration platform component.

- Project creation — identifying main project characteristics:
  - Settings:
    - Interactions inside a team or methodology,
    - Control methods,
    - Planning methods;
  - Team Selection:
    - Identifying project roles,
    - Flexible settings for each role including access rights;
  - Registration:
    - Creating an account,
    - Binding to the project,
    - Role assignment;
  - Tools:
    - Choosing tools which are necessary for the project and roles identified in the project.
- Project activities with consideration of settings:
  - Project Management System (PMS);
  - Communication: mail, chat, forums, boards, conferences;
  - Control Version System (CVS) or Subversion (SVN);
  - Issue Tracker: To-do lists and similar entities which can be strictly connected with PMS;
  - Wiki/Docs — a set of tools for document processing and storage;
  - Bug Tracker — a system allowing to record and process software errors;
  - Activities — statistics for a person, team, project, etc.
- Simulations — it is possible to simulate different situations in the software development process such as:
  - New team member,
  - Deletion of documents or/and bugs,

- Bugs or documents deleting,
- New clients' requirements.
- Consultations — accompanying educational materials:
  - Lections,
  - Recommendations,
  - Advices.

A more detailed explanation is necessary for such project setting as 'planning methods' because it is a base for calculation of control points, which are the most optimal for corrective actions.

The software development process in a simplified form can be represented by a function F:

$$F = f(T, R, P),$$

T — time constraints,

R — human resources,

P — software functional.

Software development as usual is organized iteratively. Iteration length is usually selected by an expert. In the current team collaboration platform it is possible to use special mechanism for iteration length choice.

- Iteration length is chosen by an expert (however it can be changed through the process), than within defined time constraints the most optimal requirements' sets are chosen. For each requirement a necessary team member is appointed (according to his or her role and possibilities).
- By analysis of software functions (using different representations) the most optimal iteration length is chosen for a set of functions. For a set of functions a necessary team member is appointed.
- By analysis of responsibilities and/or competences and/or personal preferences of a team member the most optimal set of functions necessary to develop is chosen. According to the set and team possibilities the most optimal iteration length is chosen.

The end of the current iteration and the beginning of the next one is the best for monitoring current project status and making corrective actions.

## 4 Conclusion

In this article only basic features of the team collaboration platform are described. By using this platform in conjunction with accompanying educational materials it is easy for students to understand what coordination within the software development process means. In perspective it is possible to extend the platform by adding a system of recommendations (based on current situation

analysis), additional tools and/or more possibilities for simulation etc.

### References:

- [1] Detienne, F. Software Design: Cognitive Aspects. — Springer-Verlag, London, 2001
- [2] Visser, W., Hoc J.-M. Expert software design strategies // Psychology of programming. — San Diego, 1990. — Academic Press San Diego, pp235-247
- [3] Crowston K., Malonet W, Lin F. Cognitive science and organizational design: A case study of computer conferencing // Human Computer Interaction. — Austin, TX, 1987. — Vol 3, pp 59-85
- [4] Belbin, R. M. Management teams: Why they succeed or fail. — Butterworth Heinemann Ltd, Oxford, 1996
- [5] Newman M. E. J., Girvan M. Finding and evaluating community structure in networks, 2004. — URL: [http://arxiv.org/PS\\_cache/cond-mat/pdf/0308/0308217v1.pdf](http://arxiv.org/PS_cache/cond-mat/pdf/0308/0308217v1.pdf)
- [6] Cockburn A. Agile Software Development: The Cooperative Game. — Addison-Wesley Professional, 2006
- [7] Malone T.W. The Interdisciplinary Study of Coordination // ACM Computing Surveys. — NY, 1994. — Vol 26
- [8] 1970. Theory of Multi-level Hierarchical Systems. With D. Macko and Yasuhiko Takahara. Academic Press.