Biuletyn WAT
Vol. LXII, Nr 1, 2013

Simulation of the IT Service and Project Management Environment

ADAM CZARNECKI, LILIANA KLICH, CEZARY ORŁOWSKI

Gdańsk University of Technology, Faculty of Management and Economics,
80-980 Gdańsk, Gabriela Narutowicza Str. 11/12
adam.czarnecki@zie.pg.gda.pl, liliana.klich@gmail.com, cezary.orlowski@zie.pg.gda.pl

Abstract. The authors of this paper present a simulation of the socio-technical system in which teams of students — regarded as low-maturity organizations — were given a task to utilize the (relatively) mature technologies that support IT project and service management. The experiment consisted of the following phases: teams formation, introduction to technologies used during the simulation, performing a set of reactive and proactive service management tasks.
Keywords: Software Project Management, IT Service Management, Maturity, ITIL, Scrum

1. Introduction

Contemporary companies cooperate with or establish their own information technology (IT) support organization. These organizations are mainly responsible for the two areas of the IT service management: service provision and service support. The development of the IT service providers relies on the maturity of their business clients and technologies that are used. These technologies should cover a variety of the IT activities: software development, project/team management or service management.

The case study of enterprises that grow organically gives the perspective of the evolution in the field of the information technology management [6, pp. 57-71]. In the process dimension it is a transition from the immature approach that lacks the awareness that the conducted activities are in fact services provided to the business, through the next phase where one can identify fragmented services, moving to the standardized services, then integrated services and, finally, reaching the level where services are being optimized. In the dimension of the service management strategy we encounter a transition from the system management through the service management, and to the business performance management as an ultimate goal of the service management plan. There is an addition to this picture: the understanding of the six layers of the service management, where the upper layers rely on these below. They include (ascending) [6, p. 85]:

1. Facilities and data centers,
2. Computer hardware,
3. Information/data,
4. Software/applications,
5. Staff/operatives,
6. Business services.

The aforementioned evolution of the IT organization can be aided with particular methods and tools, which synergetic cooperation is usually called "the technology". Naturally, the selection of the specific technology must go along with the proper context. Here, the maturity of the IT organization is the context: not only the present one, but also the desired maturity and technologies whose goal is to improve the efficiency of the team and communication between multiple teams form a wide area of the IT activities support, especially in project management.

Assuming the aforementioned remarks as a starting point of the discussion, authors of this paper would like to share some observations that have aroused as a result of the experiment carried out among the students who were grouped into cooperating teams of the IT support organizations and were equipped with the technologies that aid the IT project and service management.

The following parts of this elaboration present: environment of the study, course of the experiment, gained results and brief conclusions.

2. Study's Environment

The issue — outlined in the introduction — of reaching the maturity level at which the IT organization is able to knowingly manage services has prompted authors of this paper to conduct an experiment. This experiment's plan implied a simulation of the environment of the small IT support organization being at a certain level of the service management maturity (standardized IT services aided with technology to manage them). This organization was split into two teams: one responsible for the service operations, and the other responsible for the software development. The activities performed in the experiment served as examples to demonstrate the effectiveness of the selected technologies designed to aid the management in the areas of team working, IT services and software development.

In order to present the environment in which the experiment was carried out, one must describe its plan (2.1), teams' formation and their tasks (2.2), methods and tools applied (2.3) and typical project constraints: the scope, the schedule, and resources (2.4).

2.1. The goal and the plan of the experiment

The plan of the experiment was subordinated to the goal of the study, which was an observation of the low-mature project teams (where the low maturity means a lack of or only little experience in the IT projects along with the just emerging knowledge on technologies involved) in the environment of the selected methods and tools of the IT project and service management that are typical for the IT support organization at the certain level of maturity. The main recipients of the experiment were its participants, because supervisors wanted to transfer to students the knowledge and skills on utilizing various technologies for managing IT teams, projects and services.

The participants' list consisted of 24 students (including one of the authors of this paper, Ms. Liliana Klich) of the 3rd year of the bachelor studies in the field of the Computer Studies and Econometrics at Gdańsk University of Technology, Faculty of Management and Economics. The duration of the experiment spanned from February to May 2012 (summer semester) and was conducted as a part of the Group Project course. There were two research fellows (which are also co-authors of this paper) from the Department of Information Technology Management (Polish abbreviation used in this paper: ZZTI) to supervise the experiment.

In the course of the experiment (see: Fig. 1, abbreviations are explained later in the paper), six project teams were created. Five of them were given roles in the IT service management area and were called "service teams". The sixth team received the software development role and its purpose was to support service teams in their software change requests. Service teams — after constituting, assigning roles and getting acquainted with the technologies used in the experiment — conducted a cycle of remedial processes as a response to the request from the simulated business client pertaining to the service malfunction: starting from the helpdesk, moving to the incident management and finding the solution within the problem management process. The second set of activities of service teams referred to the change management in the business service and required an assistance of the development team. Each service team requested for a different software change that was meant to improve functionalities of the Enterprise Resource Planning (ERP) system used by the business client. This way service teams had an opportunity to perform two groups of processes: reactive (activities whose goal is to restore a full functionality of the business service) and proactive (a change that improves or creates a new business service).

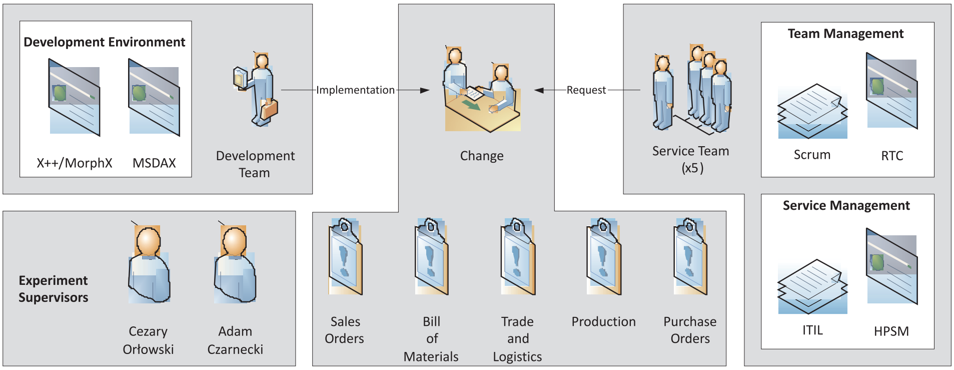

Fig. 1. Environment of the experiment (change management project)

Each team prepared a report documenting its progress during the experiment: not only listing activities, but also containing conclusions on the team's work and its cooperation with other team(s).

A more detailed description of the experiment setup and course can be found later in the paper.

2.2. Project teams

One of the first activities in the experiment was the project teams' formation. This task has been split into three stages: (1) diagnosis of every participant's predispositions to fulfill certain roles in the team, (2) designation of teams' leaders, and (3) selection of team members by leaders.

To diagnose the competencies to fulfill certain roles in the team, a test based on the theory by Meredith Belbin [1] was used: each student received a questionnaire with 56 statements grouped into 7 blocks of 8 sentences. Each participant had to allocate 10 points in each block among the sentences giving points to the sentences he or she agrees most (the number of points should reflect one's adhesion to the statement). All sentences required a self-diagnosis. Each block of statements was preceded with the following title:

1. Possible contributions to a team…
2. Weaknesses in my teamwork abilities…
3. Involvement…
4. Approach to group work…
5. I gain job satisfaction because…
6. When suddenly given a task with tight time constraints and unfamiliar people…
7. Problems I experience working within groups…

After filling in the questionnaire the points were put in the matrix that allowed grouping them into 8 columns that represent 8 different roles in the team: Completer Finisher, Co-ordinator, Implementer, Monitor Evaluator, Plant, Shaper, and Team Worker (the matrix is needed because statements and their assignment to the certain roles are intentionally scrambled in the questionnaire). As a result, every participant got a sum of points for each role — the more points the given role receives, the more such person qualifies to fulfill its duties. There are ranges of points defined for each role that assign the score to the one of four sets that describes the intensity of the role (see: Table 1).

Table 1

Belbin test — sets of roles' intensity for given ranges of points

Role in the team	Low	Medium	High	Very high
Co-ordinator	0-6	7-10	11-13	14-
Implementer	0-6	7-11	12-16	17-
Shaper	0-8	9-13	14-17	18-
Plant	0-4	5-8	9-12	13-
Resource Investigator	0-6	7-9	10-11	12-
Monitor Evaluator	0-5	6-9	10-12	13-
Team Worker	0-6	7-12	13-16	17-
Completer Finisher	0-3	4-6	7-9	10-

Results of the Belbin test among participants of the experiment have been presented in Figure 2. Each bar represents a role and its sub-bars illustrate the distribution of role's intensity level with a number of people that has reached it. Each bar sums up to 24 (which is the population of the experiment). Each participant received his/her results. They could share this information within the group.

The key parameter for the experiment supervisors was the number of "strong" Shapers, because students with the highest Shaper score were meant to become team leaders. As there were 6 teams planned (5 service teams and 1 development team), it was advisable to have 6 "very high" Shapers emerged in the test. Unfortunately, the population of the participants that reached that level of the Shaper skills was equal to 2. Therefore the supervisors have decided to select next 4 leaders from the group of people that reached 4 top results in the "high" set of the Shaper's role.

The next issue that had to be considered was the assignment for each team. It was known from the beginning of the experiment that one team would be responsible for software development activities requested by the remaining service teams. The development team had been formed before Belbin test was conducted, so it was unknown whether any Shaper would emerge among programmers. As one can

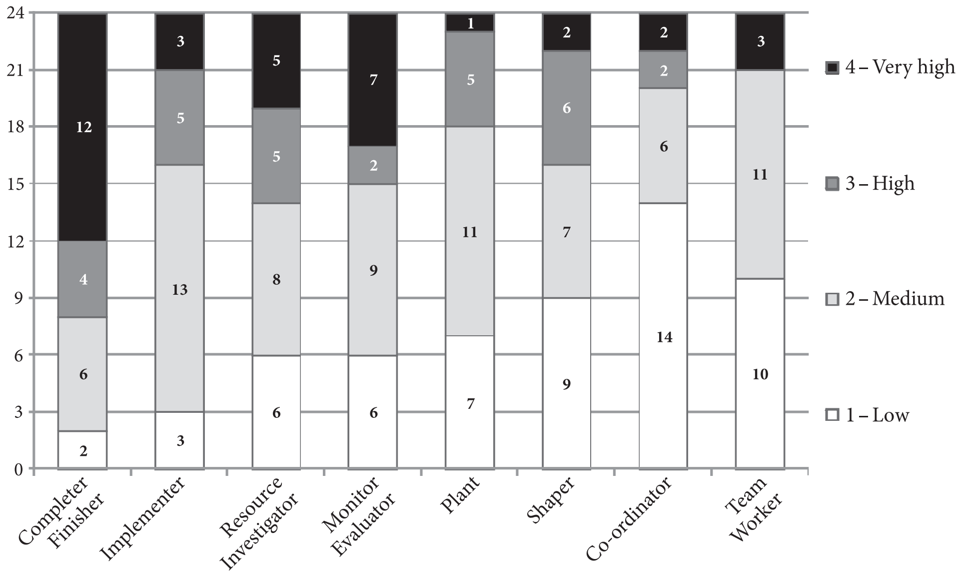

Fig. 2. Distribution of predispositions to fulfill roles in the team among participants (N = 24)

see in the Table 2, none of the development team members reached the "very high" score as a Shaper, but one person (Member 1) gained the "high" level. The remaining members of the development team reached a medium score. Therefore Member 1 was designated a leader of this team.

Table 2

Distribution of predispositions to fulfill roles in the development team (n = 4)

Role in the team	Level achieved for the given role			
	Member 1	Member 2	Member 3	Member 4
Team Worker	Low	Medium	Medium	Medium
Plant	Medium	Medium	Medium	Medium
Co-ordinator	High	Low	High	Low
Monitor Evaluator	Medium	Medium	Low	Medium
Resource Investigator	Medium	Medium	Medium	Medium
Implementer	Medium	Medium	Medium	Medium
Completer Finisher	Very high	Very high	High	Very high
Shaper	High	Medium	Medium	Medium

Leaders that have been designated on the basis of Belbin test were given a task to form their teams of 3-5 people (including themselves). They could take into account results of the competency test but it was not mandatory. After the formation of teams their members got acquainted with technologies they were going to use during the experiment.

2.3. Technologies used by project teams

There were two main groups of tasks given to service teams: (1) related to the team management within the project, (2) related to the information technology service management. The software tools and methods provided in the experiment have been presented below. They include technologies for the IT project management, the IT service management, and (3) the integrated development environment (IDE). The first two technologies were adopted by the service team while the last technology was in the area of interest of the development team. The selection of the technologies used in the experiment resulted from the research and teaching profile of the ZZTI.

2.3.1. Team management within the project

The software tool that has been intended to support team management was IBM Rational Team Concert (RTC). It is a collaboration environment dedicated to IT project teams with particular emphasis on software development projects. The client application is accessible via the Web browser.

RTC allows creating so-called project dashboards. Each team had one project dashboard created that was set up to be compatible with the Scrum method. Scrum [14] [9, pp. 189-204] is an iterative and incremental agile software development method for managing software projects. In this method, the output of the project is more important than the formal workflow. The Scrum process management of the project relies on rather short but intensive cycles called "sprints", which usually last 2-4 weeks. Tasks are registered and accounted in the backlog. Figure 3 presents a sample sprint backlog of one of the teams in the experiment.

There were two premises to choose Scrum. Firstly, choosing more formal method that has an imposed software development phases would not reflect the nature of activities in the experiment. Secondly, the freedom of defining sprints' cycles and scopes by each team has given an opportunity to trace how teams organized their operations.

Each leader received the RTC account with privileges to manage the project dashboard and he was responsible for creating user accounts of the remaining members of the team. To make the RTC more accessible, the Web client was set up at the public IP address.

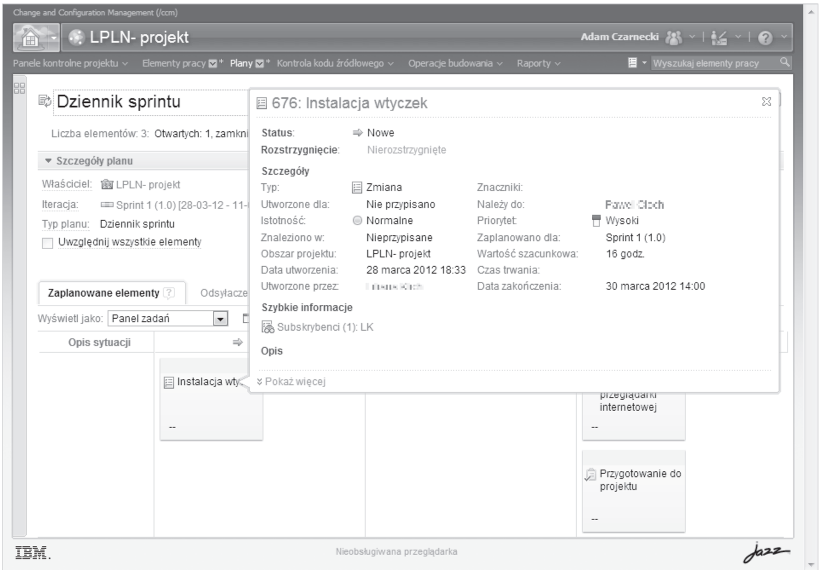

Fig. 3. RTC — sprint backlog of one of teams

2.3.2. IT service management

The IT service management module of the experiment was based on the Information Technology Infrastructure Library (ITIL), Version 3 and the supporting it software system: HP Service Manager (HPSM) 9.20.

The ITIL V3 consists of five books of the IT service management best practices, starting from the Service Strategy [7], through the Service Design [12], the Service Transition [10], the Service Operation [2], to the Continual Service Improvement [3]. Brief description of these books can be found in [4]. The scope of the experiment has been limited to the selected processes of the Service Transition and Operation: the Configuration Management, the Service Desk, the Incident Management, the Problem Management, and the Change Management.

The tool that aided teams in the proper (i.e. in conformance with ITIL V3) execution of the ITIL service management processes was HP Service Manager 9.20. This software received a Gold Level certificate from APM Group on behalf of the ITIL publisher, the Cabinet Office. It confirms not only that HPSM is compatible with this standard, but there are also a number of organizations that have successfully deployed this system. HPSM is an extensive application with a large configuration and integration capabilities. The ZZTI researchers hitherto worked mostly with standard HPSM modules and configuration ("out of the box"). Because the plan of the experiment included only basic processes, service teams did not have to modify default settings of the application and they just used predefined users/roles and processes.

HPSM offers two interfaces to users: a desktop client and a Web browser-based service (Fig. 4). The browser version is less functional than the desktop application but still sufficient for standard users. The service teams were given an access to the Web client. At the beginning of the experiment the client was only accessible in the laboratory at the university, but because of the tight schedule at the end of the semester the access via the Internet was granted, so members of teams could work from their homes.

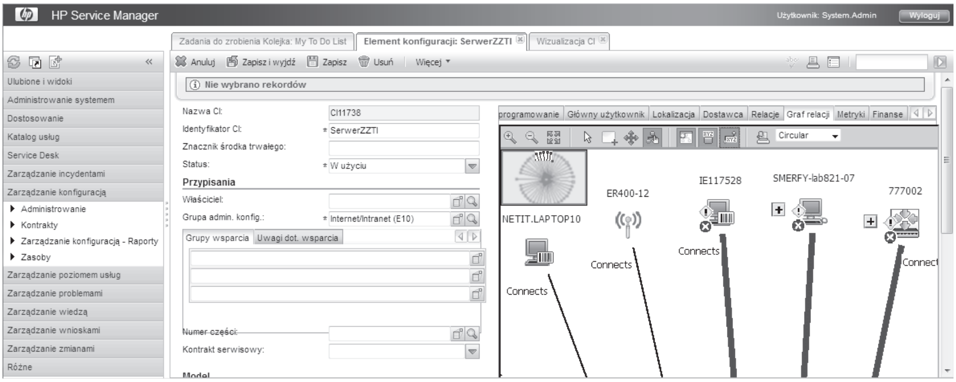

Fig. 4. HPSM — Configuration Management

2.3.3. Development environment of Microsoft Dynamics AX 2009

It was assumed in the plan of the experiment (see: 2.1) that the development team would receive change request from every service team pertaining to the business application and would implement it. It was decided by the supervisors that this business application would be the ERP class system: Microsoft Dynamics AX 2009 (MSDAX). Beside the change request, each service team was obliged to elaborate a chain of the reactive processes (see: 3.3).

There were several justifications of choosing MSDAX. Firstly, the development team was strongly interested in learning X++ (language) and MorphX (IDE) (Fig. 5). Secondly, all participants were already acquainted with the MSDAX application. Thirdly, there was an infrastructure to set up a separate virtual instance of the MSDAX servers, so the experiment would not influence the routine of the production (i.e. used in other subjects) MSDAX server.

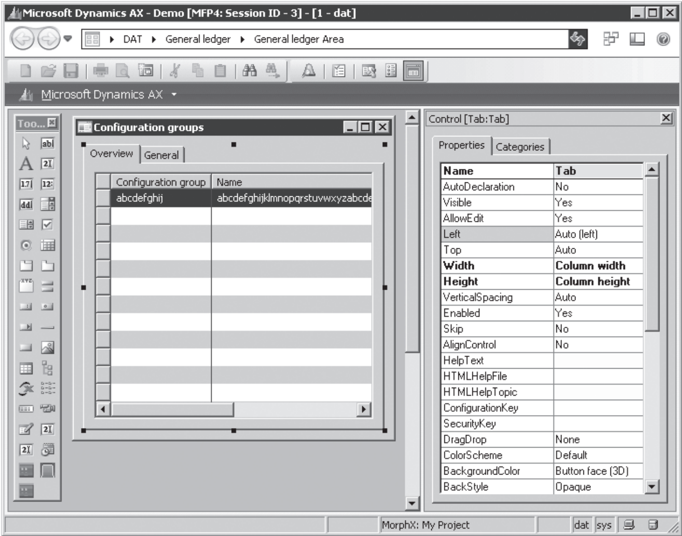

Fig. 5. Visual form editor in MSDAX

2.4. Project constraints

Project constraints are often depicted as a triangle [13, p. 11] [11, p. 14] [8, p. 716]. Each of its vertexes represents one parameter: the scope, the schedule, and resources (sometimes tailored to the financial resources and then called „the budget"). Sometimes the additional dimension is added (e.g. inside the triangle): the quality of the project's product [15, pp. 20-21]. A risk is also a frequent supplement to this model [16, pp. 19-20] [5, pp. 44-45]. Although all projects in the experiment were only simulations, it is worth — to see the bigger picture — to characterize these classic project constraints.

Each service team had to execute two projects:

1. To carry out the activities in respond to a malfunction of the business service,
2. To carry out the Change Management process that encompasses passing the request for the programming actions to the development.

The constraints described below refer to these aforementioned projects.

The **scope** of project no. 1 included receiving the complaint on the service operation by Service Desk from the business, the Incident Management, and the escalation of the incident to the problem what requires the Problem Management process to be executed in order to successfully close the problem as being solved one. The whole project should be conducted according to ITIL guidelines.

Project no. 2 included a change request pertaining to MSDAX, the deployment plan, cooperation with the development team, change acceptance, and change closure. All these Change Management phases had to conform to the ITIL standard.

Changes to MSDAX had to address one of the ten functional areas defined by the experiment supervisors (each service team had to choose one area):

1. Production (groups of products, models, dimensions),
2. Inventory Transfer Journal,
3. Accounts Receivable (orders and invoices),
4. Accounts Payable (orders and invoices),
5. Bill of materials,
6. Trade and logistics,
7. Managing offers in CRM,
8. Account management in CRM,
9. Master Planning (warehouses, min/max limits),
10. Database management (export/import).

The scope of the development team work included the change request admission from the service team, change implementation via IDE, and the service team acceptance. Such set of activities was executed five times during the experiment, for each service team change request.

Apart from conducting tasks mentioned before, each team was obliged to write the report presenting its project achievements. The structure of the report was introduced to teams in the 10^{th} week of the experiment.

The **schedule** of the experiment was limited by the duration of the semester which lasted for 14 weeks. Weeks 1-6 were dedicated to the teams preparation to the experiment (see: 3.1), while the main part of the experiment took place in weeks 7-14.

There was no budget established for the project, so the broader sense of **resources** needed to be taken into account: so-called human resources, i.e. teams described in 2.2, technologies involved (see: 2.3), and a technical infrastructure: 32 PCs running MS Windows XP/7 connected to the Internet via LAN, with a domain server (Active Directory) as a part of the authorization system. Configuration items that were a subject of the IT service activities (see: 3.2) could also be included.

All projects were only a simulation and did not have a real business client. The formal recipients of all projects were the teachers/supervisors. So, the **quality** measures were: completing the project with fulfilling the scope and on the schedule, and writing a report containing the description of team's activities along with conclusions drawn from the experiment.

The **risk** assessment was omitted during the experiment initiation, but some undesirable events happened in the course of simulation and they needed to be resolved. They are presented in Table 3.

Table 3

Risks manifested in the experiment

Constraint	Event	Cause	Severity	Resolution
Scope	Change request of the service team has not been implemented	The lack of programming experience among the development team	Medium	The service team needed to execute an alternative patch of the Change Management process of the remediation
Schedule	Delays in change requests implementation	Only one development team and five service teams requesting changes simultaneously	Medium	The development team was given a remote access to the IDE
Resources	The MSDAX production server has been disabled	The development team was gaining experience by learning from mistakes	High	The test MSDAX server has been set up; the production MSDAX server has been restored from backup

3. The course of the experiment

The description of the course of the experiment — which environment has been characterized in section 2 — should begin with the list of the preparatory steps taken with teams. One will find them in subsection 3.1. The second stage — described in subsection 3.2 — presents the Configuration Management process that each service team had to perform. Then, there are two main types of project presented: reactive processes which goal is to recover a business service (3.3) and a reactive process of the change management involving a software change in MSDAX performed by the development team (3.4).

It is worth mentioning that the most part of the experiment was taking part at weekly meetings during laboratory classes.

3.1. Preparatory steps of the experiment

The preparatory activities filled the first half of the semester and included:

1. Week 1: Belbin's competency test among students (see: 2.2),
2. Week 2:
   - Choosing leaders and giving them the task to form their teams,
   - Getting students acquainted with functionalities of Rational Team Composer and with essentials of the project management according to Scrum (see: 2.3.1),
3. Week 3: Getting students acquainted with the ITIL Configuration Management and its HP Service Manager implementation (see: 2.3.2),

4. Week 4: Getting students acquainted with ITIL processes of the Service Desk and the Incident Management along with the presentation of corresponding functionalities in HPSM,
5. Week 5: Getting students acquainted with the ITIL process of the Problem Management along with the presentation of corresponding functionalities in HPSM,
6. Week 6: Getting students acquainted with the ITIL process of the Change Management along with the presentation of corresponding functionalities in HPSM.

As one can see, the introductory activities first of all included equipping participants of the experiment with the knowledge and skills on using technologies that support service teams. Each team performed the same set of exercises that combined the theoretical knowledge on methods with skills of operating the software applications. Both supervisors put their efforts to make sure that all teams finish this prerequisite phase successfully.

While service teams were getting acquainted with the ITIL/HPSM and Scrum/RTC technologies, developers deepened the MSDAX programming technology of the X++ language and the MorphX integrated development environment.

3.2. Configuration Management

Service teams in the experiment were given a task to conduct a chain of processes in the simulated IT infrastructure of the business client to recover a malfunctioning service. But before such processes could be initiated, one had to perform a set of the Configuration Management activities in order to reflect the IT infrastructure in the HPSM Configuration Management Database (CMDB). The minimal set of the configuration, that includes only the essential items and the simplest relations,

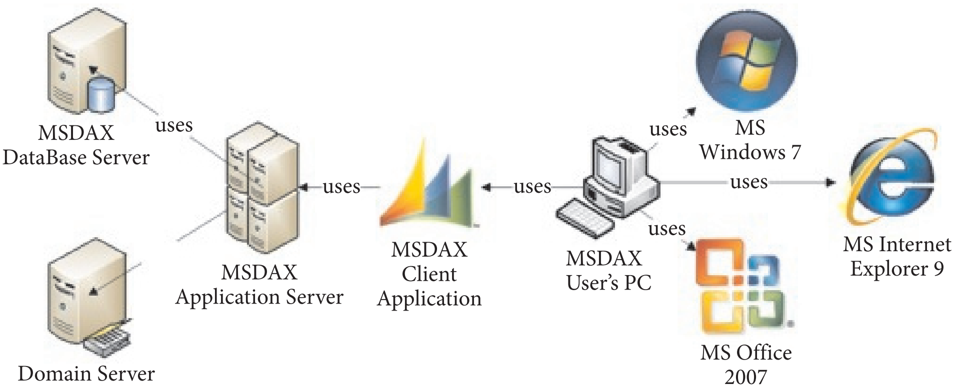

Fig. 6. A minimal set of configuration items in the experiment

is presented in Figure 6. Configuration items (CIs) records were created in HPSM along with operators' accounts: users that can be assigned as users of given CIs.

3.3. Reactive activities in the IT service operation area

The ITIL defines a reactive organization as "one which does not act unless it is prompted to do so by an external driver" [2]. In that manner we will define reactive actions as those that are triggered by such a driver. Reactive activities in the IT support organizations are often neatly called "firefighting".

Reactive activities in the experiment were planned as a chain of processes described in the ITIL Service Operation book [2]. First link of this chain is a user's interaction with the Service Desk, where the Incident Management originates as the Event. The incident is then escaladed to the problem. And the Problem Management should be the ultimate process of the simulated reactive activities. Each service team was given a task to elaborate a scenario in which such sequence would be executed in order to restore a MSDAX-based business service that the client demands. Such scenarios are presented in Table 4.

TABLE 4

A concise description of the reactive scenarios

Phase	Team 1	Team 2	Team 3	Team 4	Team 5
Event report	I can't find my company on the list	Poor visibility on the screen	When adding a product an error is displayed	Blurred application window on the mobile device used outdoors	Error while invoicing
The cause of the incident	Malfunction of the scroll bar in the application window	Direct sunlight in the room	The path do the data was changed in the developer mode	A small contrast between the text and the back-ground	Invalid account selection
Incident resolution	Use mouse roll instead of the screen bar	Rotating the monitor and/or setting up brightness and contrast	Correcting the path	Using the application indoors (ineffective)	Valid account selection
Problem resolution	Multipage instead of scrolling single page	The option to display the interface in negative	Password--protection of the developer mode	Increase of contrast by changing default colors	Moving the "supplier" field to the main table of the form
Source code changed?	Yes	Yes	No	Yes	Yes

It was a duty of the leader to define and assign tasks within the team. There were generally two approaches: (1) all the team members were collectively working on each task, (2) specialization — usually each team member was responsible for the one IT management process. This second approach required good communication among the team members.

One of the technologies to support the communication was HP Service Manager as a system that provided the workflow of tasks and information from one role to another, respecting the guidelines provided by ITIL. IBM Rational Team Concert was the second technology in use. RTC allowed teams — on the more general level — to manage tasks in their projects in compliance with the Scrum method. Both technologies were rather complementary, with only little overlapping of their functionalities. HPSM is strictly dedicated to the IT service management, while RTC serves best in supporting teams in the software development projects.

Despite the clear distinction in the scope of both technologies, there was an attempt to use RTC to support ITIL processes. Supervisors asked each service team to reflect roles and tasks of HPSM in RTC. The success was only partial: standard features of RTC allowed creating IT service roles, but not giving them any privileges specific to their duties. It was possible to create new types of project artifacts as forms (tasks etc.) with custom data fields, but developing a workflow with these artifacts was beyond the capabilities of any team. Supervisors of the experiment were not surprised by such result, for the main purpose of this attempt of technology substitution was to illustrate why two different tools are used for the project management and the IT service management.

To conclude this subsection, each service team fulfilled the scope of the project and fit within the schedule. This project did not require a Change Management or a help from the development team, however 4 of 5 service teams requested for a problem solution that required extending the project to the programming activities.

3.4. Proactive activities in the change management area

The ITIL definition of the reactive organization was given in subsection 3.3. Same source [2] also mentions the proactive organization: "A proactive organization is always looking for ways to improve the current situation". So, the proactive activities would be those in which the proactivity of the organization manifests. In the specific case of the IT support organizations it would mean initiatives that anticipate the future needs in the information technology area.

The simulation of proactive actions in this experiment was based on the ITIL Change Management [10] implemented in HP Service Manager (Fig. 7).

Each service team had a goal to devise a change in one of MSDAX functional areas (see: 2.4). Such idea of a change should be then passed to the development

Fig. 7. Standard workflow of the Change Management process in HPSM

team as a change request. After the development work is finished, the change should be evaluated and, if this evaluation is positive, the process should be closed.

As one can see (Table 5) all the requested changes applied to the MSDAX user interface. It can be explained by a rather small experience of the development team in using X++/MorphX IDE, so the team limited the scope of the requests to those which can be handled in the visual editor.

TABLE 5

A concise description of the simulated changes

Phase	Team 1	Team 2	Team 3	Team 4	Team 5
MSDAX functional area	Sales orders	Bill of materials	Trade and logistics	Production	Purchase orders
Change request	A need for the invoice no. in the sales order window	There is no quantity and unit data in the BOM constructor list	Cannot filter contractors by groups	A need for the "Dimension group" and the "Warehouse models group" data fields	A need for the "New order" button in the "General" tab
Change implemen-tation	Addition of the "VAT Invoice" data field below the "Document status" field	Addition of the button in the form and programming the click event	Addition of the "Supplier group" and the "Customer group" fields to the form grid	Placement of the above fields in the main tab of the form	Addition of the required button; proper handling the click event failed
Change evaluation	Positive	Positive	Positive	Positive	Negative

The change management project — seen from the software engineering perspective — could be an interesting case study of the software requirements specification (SRS) process. However, this issue was not included in the scope of the experiment and all the SRSs prepared by the service teams were informal and provided to the development team orally.

The cooperation phase between service teams and the development team was a bottleneck of the change management project. As it was mentioned earlier (see: Table 3) requests were queued, because the development team often worked on the change jointly. To streamline the process, a remote access to the IDE was given to developers. It did not solve the problem completely, but reduced the time needed to implement a single change: the last request in the queue awaited 3 weeks for the service.

As a result, 4 of 5 projects were completed successfully, i.e. the change requested in the SRS has been implemented in MSDAX, all Change Management phases have been documented in HPSM, and all the team/project management tasks which were beyond the ITIL scope were recorded in RTC. In one case the development team did not succeed (see: Table 5, Team 5) because of the lack of the experience. All remaining project requirements, however, were met by the service team.

4. Results of the experiment

The goal of the experiment was to observe how rather immature organizations (teams) would carry on projects that are aided with the mature information technologies consisted of software systems (RTC and HPSM) and management methods (Scrum and ITIL).

Chronologically first but intermediary result of the experiment was the formation of teams. The use of Belbin competency test gave the satisfactory effect, however just one experiment does not give any rights to claim that more spontaneous team formation would bring lesser benefits to presented projects. It just might be a hypothesis for further studies.

The opportunity to examine how teams self-organize their *modus operandi* (in the broader sense of this term) was the next achievement in the simulation. Teams mostly decided to "produce" specialists that were responsible for a specific process or, for a given process, each team member received a role with corresponding tasks. Because — as it was stated before — these teams were considered immature and newly learning methods and tools of the IT management, the use of selected technologies could greatly improve their performance.

And the use of various information technologies for managing teams and projects on the one hand, and IT services on the other, was the main axis of the experiment. Each service team could (or even: had to) complete phases of service management processes according to the ITIL, simply because HPSM forced teams to proceed in that manner. This way the following goals were achieved:

- The right sequence of activities has been assured,
- The required set of data needed to complete every phase in each process was collected using HPSM forms,
- All the data records have been stored, allowing multiple reports that may potentially be used on all levels of the IT support organization management.

A similar list of achievements can be made for the Scrum and RTC usage. This method and the tool, as a pair, allowed:

— Teams to plan their tasks,
— To assign a given task to the team member,
— To inform every team member about details of his/her duties in the project,
— To report the status of every task to the team leader,
— To analyze the collected data in the future.

To put it in straight manager terms, technologies used in the simulation were supporting (or could support) the following functions: planning, organization, motivation and control in the project.

The final result that supervisors wanted to achieve was the awareness among the participants of the experiment about certain information technologies: their pros and cons, range of application areas and types of projects the IT worker/manager may encounter.

5. Conclusions

In this paper, a simulation of the socio-technical system has been presented. A system in which virtual IT support organizations were created in order to get students acquainted with typical project activities.

During the course of the experiment its participants had an opportunity to assess the usefulness of the selected information technologies (methods and tools) dedicated to support teamwork. The simultaneous use of technologies with various functionalities gave also a chance to determine the necessity and possibility of their integration.

On the more operational/tactical level, this simulation allowed students to trace workflows in the IT service and team management, which were compatible with best practices/standards such as ITIL and Scrum. Both methods needed to be tailored to fit the scope of the experiment. Tailoring ITIL meant selecting typical Service Operation [2] processes (Incident Management and Problem Management) and one Service Transition [10] process (Change Management). Scrum, which has capabilities for managing team throughout the whole software development project, was only used by service teams as a task manager.

As a result, certain goals were achieved. Firstly, all teams completed their projects, using the ITIL/HPSM to conduct the reactive and proactive IT service management, the Scrum/RTC for the team management, and the X++/MorphX (MSDAX) for the software development. The educational goal was achieved by demonstrating — in a limited extent — the environment of IT teams' (co)operation.

A reader of this article may have at least two objections to the presented content. Firstly, the experiment described here has a stronger teaching sense than the scientific one. Secondly, due to many factors that were left out of control or which are — as it happens in the socio-technical systems — difficult to control and do not guarantee repeatability, one may think that the "case study" is the more appropriate wording than the "experiment" in naming this research. Authors recognize these both potential stipulations and regard them as a potential starting point for further studies.

This paper is based on the presentation given at the "Systems Engineering 2012" conference held by the Military University of Technology in Warsaw on September 19th, 2012.

Received November 7 2012, revised January 2013.

REFERENCES

[1] R.M. Belbin, *Management Teams*, *Why They Succeed or Fail*, Butterworth-Heinemann, Oxford, UK, 2010.

[2] D. Cannon, D. Wheeldon, *Service Operation*, Office of Government Commerce, London, UK, 2007.

[3] G. Case, G. Spalding, *Continual Service Improvement*, Office of Government Commerce, London, UK, 2007.

[4] A. Czarnecki, C. Orłowski, *Application of Ontology in the ITIL Domain*, [in:] Grzech A. et al., *Information Systems Architecture and Technology: Service Oriented Networked Systems*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2011, 99-108.

[5] M. Flasiński, *Zarządzanie projektami informatycznymi*, Wydawnictwo Naukowe PWN, Warszawa, 2006.

[6] J. Hurwitz, R. Bloor, M. Kaufman, F. Halper, *Service Management for Dummies*, Wiley Publishing, Indianapolis, USA, 2009.

[7] M. Iqbal, M. Nieves, *Service Strategy*, Office of Government Commerce, London, 2007.

[8] H. Kerzner, *Project Management. A System Approach to Planning, Scheduling and Controlling*, Tenth Edition, John Wiley and Sons, Hoboken, USA, 2009.

[9] A. Koszlajda, *Zarządzanie projektami IT. Przewodnik po metodykach*, Helion, Gliwice, 2010.

[10] S. Lacy, I. Macfarlane, *Service Transition*, Office of Government Commerce, London, UK, 2007.

[11] J.P. Lewis, *Project Planning, Scheduling & Control, A Hands-On Guide to Bringing Projects in On Time and On Budget*, *Third Edition*, McGraw-Hill, New York, USA, 2001.

[12] V. Lloyd, C. Rudd, *Service Design*, Office of Government Commerce, London, UK, 2007.

[13] S.E. Portny, *Project Management for Dummies*, *3rd Edition*, Wiley Publishing, Indianapolis, USA, 2010.

[14] K. Schwaber, J. Sutherland, *The Scrum Guide*, http://www.scrum.org/scrumguides/ (access: 2012-07-19).

[15] S. Snedaker, *How to Cheat at IT Project Management*, Syngres, Rockland, USA, 2005.

[16] K. Waćkowski, J.M. Chmielewski, *Wspomaganie zarządzania projektami informatycznymi. Poradnik dla menedżerów*, Helion, Gliwice, 2007.

A. CZARNECKI, L. KLICH, C. ORŁOWSKI

Symulacja środowiska zespołowego zarządzania usługami i przedsięwzięciami informatycznymi

Abstract. W artykule zaprezentowano symulację systemu społeczno-technicznego, w którym zespołom złożonym ze studentów — traktowanym jako organizacje o niskiej dojrzałości — powierzono zadanie wykorzystania stosunkowo dojrzałych technologii wspierających zarządzanie przedsięwzięciami i usługami informatycznymi. Na eksperyment składały się następujące etapy: sformowanie zespołów, wdrożenie ich w technologie użyte w symulacji, wykonanie zestawu działań reaktywnych i proaktywnych z zakresu zarządzania usługami.

Słowa kluczowe: dojrzałość, ITIL, Scrum, zarządzanie przedsięwzięciami informatycznymi, zarządzanie usługami informatycznymi