УДК 004.4

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ ГНУЧКИХ МЕТОДОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ПРАКТИЦІ СУЧАСНИХ ПІДПРИЄМСТВ

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ИСПОЛЬЗОВАНИЯ ГИБКИХ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРАКТИКЕ СОВРЕМЕННЫХ КОМПАНИЙ

FLEXIBLE METHODOLOGIES EFFICIENCY RESEARCH IN PRACTICE OF MODERN SOFTWARE DEVELOPMENT COMPANIES

Н.Д. РУДНИЧЕНКО, канд.техн.наук
Одесский национальный морской университет, Украина

Наведено результати дослідження ефективності використання гнучких методологій розробки програмного забезпечення в практиці сучасних компаній. Викладено результати проведеного опитування щодо використання методологій управління проектами, а також результати вартісного аналізу 9 різних проектів. Виявлено особливості різних гнучких методологій розробки при їх практичному застосуванні.

***Ключові слова**: гнучкі методології розробки програмного забезпечення, вартісний аналіз проектів.*

Приведены результаты исследования эффективности использования гибких методологий разработки программного обеспечения в практике современных компаний. Изложены результаты проведенного опроса по использованию методологий управления проектами, а также результаты стоимостного анализа 9 различных проектов. Выявлены особенности различных гибких методологий разработки при их практическом применении.

***Ключевые слова:** гибкие методологии разработки программного обеспечения, стоимостной анализ проектов.*

The results of research on the effectiveness of the use of flexible software development methodologies in the practice of modern companies are presented. The results of the survey on the use of project management methodologies, as well as the results of the cost analysis of 9 different projects are presented. The peculiarities of various agile development methodologies are revealed at their practical application.

***Keywords**: agile software development methodologies, project's cost analysis.*

Introduction. Modern trends in the development of methods for the effective management of software development in the domestic and foreign markets indicate the growing relevance of the use of flexible approaches (agile) in practice [1]. The main problem in the management of projects in the field of information technology is to perform the tasks in a short time with high quality and a budget acceptable to the customer (triple bound) [2].

Software development, like any other technical discipline, deals with the following main problems: quality, cost and reliability [3]. In this regard, the proper organization of the software development process is the basis for achieving the planned result in the expected time, with the expected level of quality and with an adequate budget value [4]. Among the common problems of the software development process are the following [5]:

1. Change requirements directly in the development process.
2. Fuzzy distribution of responsibility for the work performed and its result between team members.
3. The presence of a continuous stream of small, «fast», piling up demands, distracting developers and managers from the main line of work.
4. As a consequence, failure to meet deadlines, inflation of budgets, loss of work quality.

Agile does not include specific practices, but determines the values and principles by which successful teams are guided. Most flexible methodologies are aimed at minimizing risks by reducing development to a series of short cycles, called iterations, which usually last one to two weeks. Each iteration itself looks like a software project in miniature and includes all the tasks necessary for issuing a mini-growth on functionality: planning, requirements analysis, design, coding, testing and documentation [6].

Relevance. Modern companies often implement existing and proprietary practices in managing the software development process, relying solely on experience on existing cases and management preferences [7-9]. This approach is not always very effective because it doesn't include the stages of project risk assessment and cost analysis in order to find the best methodology for the particular project development [10; 11].

Goal of the work. The aim is to study the effectiveness of using various Agile-methodologies in practice by conducting a cost analysis of planned and actual costs taking into account project risks based on statistical data collection.

Basis of material. To collect statistical data, a targeted interview was developed. It consists of 11 questions, 9 questions containing the answers, and the remaining 2 questions are aimed at a creative text response that reflects the respondent's private opinion. The interview's target audience consists of the middle and top level managers of software development units, team leads, project managers and providers of practical integration Agile-techniques (Scrum-masters, etc.). The interview was created and placed in the Internet using Google Forms.

The results of the survey on the software development methodologies used and on the most critical design risks are shown in Fig. 1 and Fig. 2. As can be seen in the above figures, the most frequently used methodology in practice by the respondents is Scrum, and the most critical risks are errors in estimating the scope of work and the human factor.

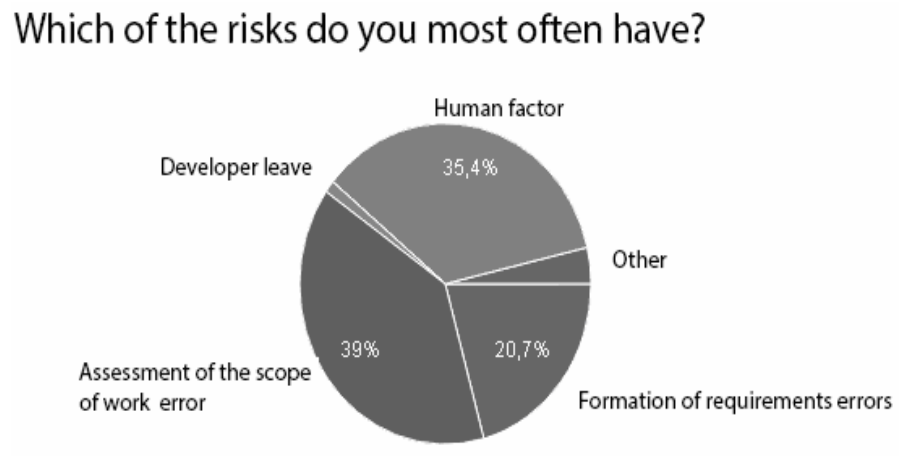

Fig.1. The most critical design risks

For a more accurate assessment of the effectiveness of the methodologies, 9 different cases were considered, for each of them a cost analysis of project was carried out. The information on the projects was received from a different development teams and is insider trading.

The cost analysis of the project (C) was defined as the product of the project execution time (T) by the cost of the team (K). T was defined as the ratio of the project volume in the story points (W) to the average team speed (S) (the average number of tasks per sprint).

The carried out cost analysis of the reviewed projects allowed to establish the planned, actual and final (taking into account the risks of exceeding the budget, external factors and delinquency) value. Summary results of the cost analysis of the 9 reviewed projects are shown in the table 1.

In most of the cases examined, the risks of exceeding the budget were realized, and in projects № 4 and 7 there was also a delay. The inherent risks of not timely delivery of results for projects 4 and 7 were about 10 %, which could not compensate for the total budget excess. This is due to the specifics of the Waterfall development methodology.

Analyzing all 9 considered projects, it should be noted that in practice, large values of risks are laid by teams numbering more than 6 people, with the lowest risk values being laid with Waterfall (5-10 %), more significant with Scrum (10-20 %), the most Large (from 30 %) for XP / TDD. High risks, in comparison with others, for XP are related to the specifics of the methodology itself.

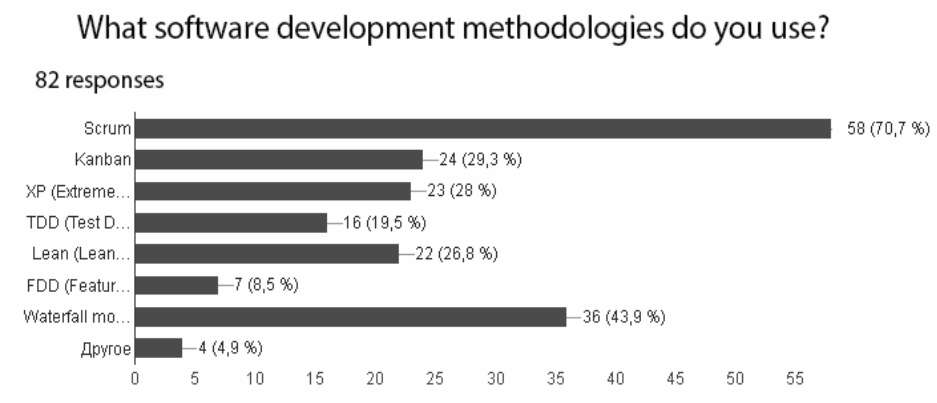

Fig. 2. Methodology usage diagram

With an increased degree of strain on the development team (lengthening the working day, consolidating the list of tasks to be carried out on the sprint, reinforced monitoring, nervous conditions due to tight deadlines).

When the obtained values of cost estimates for all projects are aggregated and averaged, it can be argued that the most time-consuming, technically complex, costly and risky in terms of probability to exceed the deadlines are web-systems and web-applications projects. Of the cases examined, this is № 2, 3, 4.

This is largely due to the necessity to use automated means of checking and testing the work of the product on iterations, allocating funds for the acquisition and deployment of servers for continuous integration, payment of manual and automated testers, and the difficulty of fully integrating solutions on hos-ting.

In this regard, in the projects of this category of software products, it is necessary to pawn large risks of exceeding the budget and delay. In the reviewed cases the budget for these projects is greater than other types. The final cost more often than the planned, often possible delay.

Less risky projects are mobile applications with a single or two-tier architecture, i.e. Without the use of application servers.

In our case, an example of such a project is № 1. According to the data, the size of the team for such projects can be less voluminous than for web development, which, on the one hand, reduces the risks of exceeding the budget, and on the other hand increases the risk of loss of qualified specialists.

The threshold of entry into the field of mobile development is higher than in the web and the number of job seekers for these vacancies is much lower than in the field of web development. Nevertheless, the project number 1 indicates a possible exceeding of the budget in practice for the reasons described above. The total cost, in about half the cases, may exceed the planned value due to risks.

Table 1

Summary results of the cost analysis of the reviewed projects

№	Metho-dology	Team Size (man)	Planned project's duration (days)	Actual project's duration (days)	Planned project's budget (UAH)	Actual project's budget (UAH)
1	Srum	8	30	30	1 800 000	2 120 000
2	XP + TDD	4	15	15	416 000	416 000
3	FDD	8	30	30	3 700 000	3 680 000
4	Water-fall + Agile	4	75	82	349 700	380 000
5	Water-flall	7	89	89	10 000 000	12 000 000
6	Scrum	9	53	53	10 000 550	10 000 000
7	Water-fall	4	35	85	336 000	1 000 000
8	Scrum	3	45	45	360 000	360 000
9	XP	3	45	45	450 000	450 000

The least risky, from the point of view of the failure of terms, are desktop projects, most of which, within the framework of the projects reviewed, was performed on the basis of Agile-methodologies Scrum and XP (№ 6, 8, 9).

This is explained by the fact that the time and material costs for testing and deployment of such solutions are less than for web or mobile applications, and modern tools for cross-platform development (Java, Python, etc.) allow developers to reduce the amount of work required to integrate the final solutions for different operating systems and platform.

Risks of delay, respectively, are also lower. The final cost usually coincides with the planned one.

Extrapolating the obtained data, it should be noted that different agile practices and their combinations are appropriate for different cases.

We should notate that the general project's classification according to the types described above: web applications, mobile applications and desktop applications. Also, should be classified and the team in size: large (more than 10 people), consisting of individual subgroups; medium (5-9 people) and small (up to 5 people).

Depending on the specifics of the project, different methodologies will give different technical and economic development effects. In particular, XP / TDD will allow us to increase the speed, but most likely reduce the quality.

Scrum will improve the quality, but will reduce the speed and increase the budget. FDD is less predictable, because on the one hand, it can raise the budget, on the other hand it can reduce the risks of delinquency.

Thus, taking into account the various risks, the most feasible from the financial point of view among the cases considered are desktop projects, next comes mobile applications and web systems.

It is reasonable to assume that with the increasing team's size it is reasonable to use agile-methodologies with more centralized control and monitoring mechanisms, which can improve the overall efficiency and productivity of the work.

At the same time, it is important to ensure that these practices do not develop into a Waterfall model, characterized by an excessive degree of bureaucracy, not transparency and formalization, which, as practice shows, leads to exceeding the deadlines and budget of the project.

For case studies on the basis of Scrum, almost complete coincidence of the planned and actual budget was found (discrepancy even less than 5 %). The most significant excess of the budget occurred when using the methodology of waterfall (more than 200 %).

It allows to affirm that Agile methodologies are more effective than Waterfall. This is due to the flexibility of the organization of communication between the client and the team, the absence of bureaucratic delays and instances, the reduction of the degree of formality of labor control for developers and the holding of regular internal meetings (meetings).

Conclusions. The conducted researches approved the dominating popularity of the Scrum methodology in the practice of software development.

This is due to the fact that it supports more tools to maximize the flexibility of the software development process and has a smaller integration threshold in practice, compared to XP, TDD, FDD.

Also it has such advantages for the team: it is easier to interact; it is necessary to spend less time and effort on the implementation of refactoring and drawing up project-reporting documentation; it is easier to understand the state of the project by holding short daily Scrum-meetings; it allows to summing up the results with the formation of recommendations on improving the work on subsequent projects through retrospective analysis.

However, the values of the risks for exceeding project's deadlines for Scrum methodology are quite high.

Their minimization is possible through the implementation of preventive modeling of possible scenarios of the situation evolution on the basis of Monte Carlo methods, fuzzy logic and regression prediction.

REFERENCES

1. *Амблер С. Гибкие технологии: экстремальное программирование и унифицированный процесс разработки / С. Амблер. – СПб.: Питер, 2012. – 416 с.*
2. *Мельченко А.А. Гибкие методологии разработки программного обеспечения: экстремальное программирование управления проектами / А.А. Мельченко // Известия Гомельского государственного университета имени Ф. Скорины. – 2015. – № 2 (89). – С.146-150.*
3. *Карпов Д.В. Гибкая методология разработки программного обеспечения / Д.В. Карпов // Вестник Нижегородского университета им. Н.И. Лобачевского. – 2011. – № 3 (2). – С. 227-230.*
4. *Баронов В.В. Информационные технологии и управление предприятием / В.В. Баронов, Г.Н. Калянов, И.Н. Титовский. – М.: АйТи, 2004. – 512 с.*
5. *Карминский А.М. Информатизация бизнеса / А.М. Карминский, В.П. Нестеров, Б.В. Черников. – М.: Финансы и статистика, 2004. – 351 с.*
6. *Книберг Х. Scrum и XP. Заметки с передовой / Х. Книберг. – М.: Enterprise Software Development Series, 2015. – 94 с.*
7. *Кон М. Scrum: гибкая разработка ПО / М. Кон. – М.: Вильямс, 2011. – 576 с.*
8. *Криспин Л. Гибкое тестирование. Практическое руководство для тестировщиков ПО и гибких команд / Л. Криспин, Д. Грегори. – М.: ООО И.Д. Вильямс, 2010. – 464 с.*
9. *Мартин Р. Быстрая разработка программ. Принципы, примеры, практика / Р.С. Мартин, Д.В. Ньюкирк, Р.С. Косс. – М.: Вильямс, 2004. – 752 с.*
10. *Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. – СПб.: Питер, 2002. – 496 с.*
11. *Highsmith J. Agile Software Development Ecosystems / J. Highsmith. – Addison-Wesley Professional, 2002. – 326 p.*

Стаття надійшла до редакції 15.09.2017