Wastage and Its Elimination in Software Development Projects in Europe in the Context of Lean IT

Piotr Sycz
University of Gdańsk
Gdańsk *piotr.sycz@ug.edu.pl*

Elżbieta Wojnicka-Sycz
University of Gdańsk & Gdańsk University of Technology
Gdańsk *elzbieta.wojnicka- sycz@ug.edu.pl*

Abstract

The article introduces the concept of wastage and its elimination in software development projects in the context of Lean IT. Empirical research in 142 European computer programming companies shows the types of losses and their causes, as well as the techniques used to minimise them. The main losses are delays and waiting, unnecessary meetings and switching between tasks. The most common causes of losses are overly ambitious client requirements and running multiple projects at the same time. Techniques used to reduce wastage include agile project management and proper planning. Larger organisations tend to have higher levels of wastage. Statistical analysis has shown that delays are associated with unnecessary processes and movements. Poor communication in particular causes various types of wastage, while software defects in particular have multiple causes. Logistic regressions showed that the elimination of wastage correlates with a better competitive position and development prospects of companies.

Keywords: wastage, software development, project management, Lean IT, Europe.

1. Introduction

Software development projects (SDP) typically face implementation problems. According to the Chaos 2020 Report's database of 50,000 projects, only 26% of software development projects developed from scratch, and 37% of projects developed using components run smoothly [1]. To increase the efficiency of software development projects, agile project management methods are increasingly being used. These are based, among other things, on iterative development and the ability to make changes to the project [2]. There are also project management methodologies based on the industrial practice of the Toyota Production System (TPS), such as Lean IT or Le-agile, which combine Lean IT and agile methodologies [3, 4].

The Lean concept adapted to the specific work of IT companies was first demonstrated in Mary and Tom Poppendieck's (2003) 'Lean Software Development: An Agile Toolkit for Software Development Managers'. Lean IT is a set of practices that slims down structures and streamlines processes. Lean philosophy describes the 'what' to do to reduce wastage, while Agile is a way of eliminating low value-added activities [5]. Lean Thinking is about identifying and eliminating losses in a value stream. The process includes 1. activities that clearly create value; 2. activities that do not create value for the customer but are necessary; and activities that do not create value for the customer and are unnecessary and should be removed - wastage [6].

Johnson & Mulder [1] show that Flow-like micro-projects based on Lean IT principles have the highest success rate - 71%. Flow modernisation reduces decision latency because it is based on self-directed teams that can make their own decisions. ‘Infinite Flow’ method of software development increases customer satisfaction. Flow-like projects generate higher net value because the improved software functionality is delivered quickly in small

portions. Infinite Flow reduces overheads to 20% compared to 80% in waterfall projects and 52% in agile projects [1]. Infinite Flow has no budgets or project plans. There is a budget for the direct costs of software development. There is also the cost of managing the pipeline of activities, but with greatly reduced project overheads. This is achieved by reducing and eliminating most of the current project management activities. Changes to the project occur continuously, but in small steps that are useful and more acceptable to users. It means a very 'lean' implementation of software development projects with the elimination of non-value adding activities [1,4].

On 5 April 2024 we found 221 articles in the Scopus database on IT/ICT/software development projects or project management and lean, waste, muda, leagile, lean IT, flow or the related term 'kanban' in the Business and Management and Computer Science records. We then analysed the abstracts of all these articles, from which the 68 most relevant to the topic of the article were selected, and in-depth full text analysis was carried out on 27 articles. The issue of loss types in IT alone was addressed in only five articles [6, 7, 8, 9, 10]. Other articles analysed ways to increase the efficiency of projects using concepts related to lean thinking such as [11, 12, 13, 14, 15, 16, 17]. Although the use of Lean to improve the management of software development projects has been explored by academics and practitioners since the publication of Mary and Tom Poppendik in 2003, it should be noted that there is still little academic literature and rare research on the subject. This paper attempts to fill this gap by presenting the results of a survey on the extent and types of wastage in IT projects in software development companies, that is, in the NACE[1] Computer Programming division from Europe, together with techniques to reduce it. In addition, we explored the relationships between various issues raised in the interviews to gain a deeper insight into the causes and effects of wastage in software development projects. Specifically, the following hypotheses were tested using statistical and econometric methods: H1. Different types of wastage are associated with each other, H2. Different types of wastage causes are associated with specific types of losses, H3. Wastage is associated with the scale of SDP overruns in terms of schedule, budget and scope and affects the competitive position and growth prospects of companies.

2. Literature review: wastage in the context of Lean IT

The Toyota Production System (TPS) originally defined the following types of wastage: overproduction, waiting, transport and movement, extra processing, inventory, defects, wasted talent [6] [18]. In the context of software development projects, losses include extra features, delays and waiting, handovers and partially done work, defects and failure, context and task switching, unnecessary movement, extra processes and unused employee creativity [9, p.203] [15, p.0662]. Wastage is "any human activity that consumes resources but does not create value" [19, p.766; 20, p.15]. In IT organisations, wastage is of a different nature than in manufacturing organisations. Examples of losses in IT organisations are: overprovision of services, over-specification (functionality that is not used in the final product), re-processing of work or data, software bugs, waiting for approval, handling defective software. Value is utility that has been delivered to the customer at the right time and at the right price. For example, value to the customer in terms of software development is the performance of service contracts at the right level and the reliability of the software [19] [4].

Sedano et al. in [6] identified nine types of wastage in software development: "building the wrong feature or product, mismanaging the backlog, rework, unnecessarily complex solutions, extraneous cognitive load, psychological distress, waiting/multi-tasking, knowledge loss, ineffective communication". Sallin et al. in [8] in their research, based on the responses of 26 IT specialists from a large IT company (out of 164 who received the questionnaire), used the following categories of waste for self-reporting "Building the wrong feature or product", "Mismanaging the backlog", "Rework", "Unnecessarily complex solutions", “Extraneous cognitive load” "Psychological distress", "Waiting/multi-tasking", "Knowledge loss", "Ineffective communication", "Management & organisational aspect", "Manual work", "Other tasks". The results showed that typically

[1] Nomenclature statistique des Activités économiques dans la Communauté Européenne

4.6 hours per day were wasted. The most significant waste was rework, followed by other tasks such as administrative demands. In a survey of 80 IT experts in Poland, the categories of losses in software development projects were ranked according to the most frequently mentioned as follows: 1. waiting, delays in projects; 2. defects; 3. unnecessary employee effort, switching between tasks; 4. unused employee creativity; 5. overproduction - unnecessary features; 6. inventory, partially done work that is not fully used; 7. unnecessary, too frequent meetings; 8. unnecessary processing, training in unused technologies [4].

There are five Lean principles that can also be transferred to IT organisations [19, p.766; 20]. These principles are: value, value stream, flow, pull and perfection. Value is the definition of value from the customer's perspective, Value stream is the identification of all the steps necessary to design, order and manufacture a product throughout the value stream to remove the various types of waste that occur. Flow is the creation of a continuous flow without interruption. Pull is producing only what the customer orders. Perfection is the elimination of successive layers of wastage as they are discovered. Lean principles relevant to the software development are such as waste elimination, built-in quality, knowledge creation, addressing bottlenecks, proper rewarding system, continuous improvement in the form of Kaizen, pull instead of push system, using Kanban boards, control of Work in Progress, mapping value stream, levelling the workload, optimising the whole, increasing flow, reduction of friction, usage of fast feedback loops, understanding behaviour of queues, decentralised control, exploitation of variability, reduction of batch size, making process policies explicit, usage of models to recognise opportunities of improvement [15] [17].

An experiment conducted by Middleton & Joyce [21] showed that the application of Lean management practices increased the efficiency of a project following the Agile methodology. Lean tools such as Kanban boards were used and only tasks placed on the board were completed, work-in-progress limits were introduced, and bottlenecks were analysed during stand-up meetings. In addition, quality was improved by monthly data analysis to summarise imperfections and draw conclusions. Improvements in efficiency were achieved by using the pull method, that is, only tasks with the highest value for the customer were implemented. Kaizen - continuous improvement was based on data, not job descriptions. There was no project manager or scrum master. All team members collaborated to remove bottlenecks.

Many tools known from the Lean concept for manufacturing, but also from other management concepts such as Total Quality Management (TQM), can be applied to IT management. You can use problem trees or Ishikawa diagrams, the Deming cycle (Plan-Do-Check-Act), the 5S method for organising and keeping the workplace in order (Sort, Set in Order, Shine, Standardize and Sustain[2]). "Gemba walks", meaning the manager being where the added value is created is crucial in agile management in IT. The 'Ask 5 times why' method is an effective analytical tool for identifying the root causes of a problem. The question is asked five times based on the information received in the previous answer until a conclusion is reached [22], [23],[4]. Aldaeej et al. in [11] proposed a lean method of technological debt management based on sprint velocity. Monserrat et al. [12] showed examples of using lean tools such as value stream, tact time, Andon cord or rapid prototyping with short iterations to deliver products quicker to customers with noticing bugs earlier. A diagram for estimating task lead times in IT projects was proposed by Weflen et al. in [13].

Flow methods of software project management, derived from Lean practice, encourage visualisation of work as it passes through different states from Planned to Done, collaboration, value measurement and knowledge sharing. Flow focuses on managing queues using a cumulative flow diagram [16]. To create a smooth flow of work, Lean recommends smaller work packages, frequent builds and prioritisation of work. Smaller work packages help to better balance the workload across team members [24].Gunawan & Budiardjo in [14] gave recommendations for software process improvement in small IT

[2] https://www.epa.gov/sustainability/lean-thinking-and-methods-5s, Accessed 13.04.24

companies using practices related to lean philosophy such as Kanban and DevOps. These recommendations are: standardisation and coordination of processes, mandatory testing of each artefact and generation of documentation in open format, adoption of Scrum-style stand-up meetings, standardised software process workflow, product entry lists including traceability record and risk register, standardisation of the list of work products, mandatory change request document with the project manager as its owner, key performance indicator for the project manager being the verification process, use of responsibility matrix, project plan with a list of minimum contents, attendance as well as agreement on roles. Sycz in [4] proposed a model of proactive wastage detection and elimination in the form of a diagnostic-creative procedure following the cycle of D-diagnose, A-analyse, P-plan, I-implement, C-create, E-evaluate. The procedure uses management methods such as the problem tree with elements of the Ishikawa diagram, the brainstorming method, indicator analysis, the ABC method and the Eisenhower matrix. A list of ready-made solutions was also proposed, including various wastage reduction methods for specific losses. These methods embrace such as Pull, Visualisation, Value Stream Mapping, Demming Cycle, Gamification, Kanban Board and others. In the analytical and diagnostic phase, the author proposes a questionnaire, also used in this research, which can help companies to diagnose the wastage present in their projects and its causes.

3. Data and Methods

In 2024 we conducted a Computer Assisted Web Interview (CAWI) with computer programming companies registered in the Orbis database[3] in Europe, the USA and Canada (around 30 thousand e-mails were sent). We only received responses from 142 European companies. The questionnaires were sent twice, on 23-28 December 2023 and on 19 January 2024. The questionnaires were sent by the IT Centre of the University of Gdansk. Questionnaires were sent to companies in each country in both English and the native language, except for Poland, where we sent questionnaires in Polish only. In total, we sent questionnaires to 18858 companies from the countries from which we received answeres. We received responses from Central and Eastern European (CEE) countries: Poland (40), Estonia (8), Hungary (7), Lithuania (6), Bulgaria (4), Slovenia (3), Slovakia (3), Romania (2) and the Czech Republic (1), and Western European (WE) countries: Portugal (20), Finland (20), Germany (17), United Kingdom (6) and Norway (5). Questionnaires were also sent to companies in Denmark, France, Latvia, Belgium, the USA and Canada, but we received no replies. For other European countries in the Orbis database, very few or no companies were found with e-mails and number of employees and at least one record of recent financial data, so they were not included in the survey.

We asked respondents the following questions about waste, where they could mark all possible options and add one more:

1. What type of waste is most common in your company's software development projects? Unnecessary meetings, Switching between tasks, Losses due to software bugs/defects; Unnecessary movement of people, work, knowledge hindering smooth execution of tasks; Over-quality - unnecessarily added features, Too much Work in Progress (WIP), Untapped creativity of employees, Delegation of work to other teams or individuals, Unnecessarily implemented processes, Losses that may occur due to the use of outdated technology, Delays and waiting.

2. What are the main causes of waste in the implementation of IT projects in the company? Poor communication within the project team, Poorly diagnosed customer requirements, Overly ambitious schedule, Execution of several projects simultaneously, Poor distribution of tasks among project team members, Lack of effective monitoring and control, Overly ambitious customer requirements, Excessive tasks of the project manager, Too small project budget, Insufficient staff qualifications, Technology.

3. What techniques do you use to reduce waste in projects? Agile project management (kanban, scrum, product increment), Proper planning of time and tasks (realistic schedules and deadlines), Better communication (e.g., inter-team meetings, frequent communication,

[3] https://login.bvdinfo.com/R0/Orbis

especially in problem situations), Supervision by experienced employees, Monitoring of the current workload of employees (use of ticketing), Use of project management software, Better recruitment of suitable employees, Pair programming, Proactive detection of waste.

We also asked what percentage of the organisation's software development projects were significantly over budget, over schedule or over scope. In addition, we asked what percentage of the organisation's software development project activity, on average, can be considered wasteful and does not add value to the external or internal customer (the organisation's employees). We asked about the size of the organisation in terms of employees: micro - up to 9 employees, small - 10 to 49 employees, medium - 50 to 249 employees and large - 250 employees and more. As only 7 respondents from large companies replied, we analysed them together with the medium-sized companies. We also asked about the assessment of the company's position in the industry in the country (strong, average and weak) and the assessment of the company's development prospects in terms of revenue or profit in the next year (very good, good, average, weak, very weak). Respondents also indicated their role in the company (CEO/director/owner, IT project manager, IT specialist and other). We also asked whether the companies had foreign customers (yes, no, hard to say). As not all respondents answered each question, the exact number of responses collected is always given in the presentation of the results.

We performed a statistical analysis of the significance of the differences in the proportions between the groups in terms of location and enterprise size. We used the two-sample test for large samples in the Stata package, depending on the proportion and size of the population in question [25]. In addition, contingency tables were created showing the co-occurrence of different types of wastage, as well as types of losses and their causes. Pearson's chi-square test was used. Pearson's Chi-square test measures whether the observed values equal the expected value, as claimed by the null hypothesis. Its rejection means that the observed values are significantly different from the expected value, and hence there is a relationship between the variables due to their co-occurrence [4].

In addition, we estimated models using logistic regression to assess which characteristics of the surveyed companies (number of causes of wastage, number of wastage reduction techniques implemented, company size, location) increased the likelihood of having good development prospects or a strong competitive position in the industry. An odds ratio was also calculated to indicate how much more likely it is that a company with a given characteristic will stand out in terms of competitive position and development prospects [26].

4. Results

115 respondents from the CP companies surveyed were CEOs, directors or owners. Some respondents indicated specific roles such as finance director, sales director, chief of staff, department head, senior management. 13 respondents were IT project managers and 11 IT specialists, while 3 respondents indicated another role. 52.5% respondents declared a low proportion of projects that significantly exceed budget, schedule or scope (Fig. 1, Table 1). More than half of software development projects were problematic for 30% of European computer programming organisations. No significant differences in project efficiency were observed between the former communist countries of Central and Eastern Europe and Western European countries. The CEE countries stood out only in terms of the highest proportion of problematic projects. Slightly higher proportions of the most problematic projects and lower proportions of the least problematic projects were observed for small enterprises. This suggests that there may be a problem of scaling up businesses while keeping wastage at a low level.

Almost 39% of respondents said that less than 10% of the activities in their software development projects can be considered as wastage. For about 49% of respondents, waste is greater than 10% but less than 30% (Fig. 1, Table 1). This represents a satisfactory level of waste elimination, which may be due to the increasing popularity of Lean IT and similar

[4] http://manuals.pqstat.pl/statpqpl:porown3grpl:nparpl:anova_chipl, Accessed 6.04.2024

methods in software development projects. In Western European countries, no respondents reported more than 50% wastage. The percentage of respondents from WE reporting the lowest percentage of wasteful activities was higher than in CEE. It appears that wastage reduction is a higher priority for IT companies in the WE countries, which may be related to the generally higher labour costs there.

The majority of micro enterprises reported the lowest rate of waste, which was significantly higher than for small, medium and large enterprises. Large enterprises have a higher administrative burden than micro enterprises, but are able to carry out larger projects. However, small, medium and large enterprises were more likely to report a strong position in industry than micro enterprises.

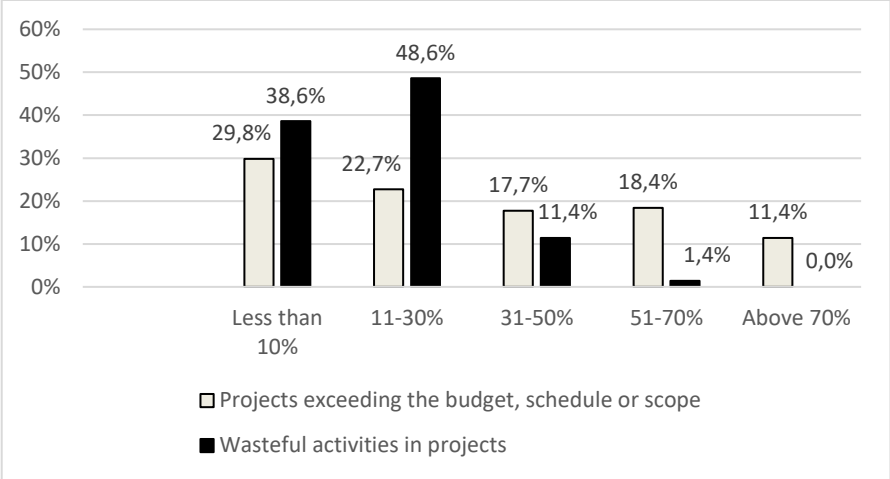

Fig. 1. Share of projects exceeding budget, schedule or scope and the share of wasteful activities in software development projects according to responents from Europe (percentage of respondents). Source: the survey.

Table 1. Percentage of company's software development projects significantly exceeding the budget, schedule or scope and the percentage of wasteful activities in projects.

	Projects exceeding the budget, schedule or scope					**Wasteful activities in projects**				
Category	**Location**		**Number of employees**			**Location**		**Number of employees**		
	CEE	**WE**	**50 and more**	**10-49**	**Less than 9**	**CEE**	**WE**	**50 and more**	**10-49**	**Less than 9**
Less than 10%	27%	32.8%	30.8%	24.1%*	*35.6%*	*33.3%*	44.1%*	29%**	35.1%**	*51.1%*
11-30%	25.7%	19.4%	20.5%	24.1%	22.2%	52.8%	44.1%	52.6%*	52.6%*	*37.8%*
31-50%	17.6%	17.9%	20.5%	20.7%*	*11.1%*	11.1%	11.8%	10.5%	12.3%	11.1%
51-70%	16.2%	20.9%	18%	12.1%*	*24.4%*	2.8%	0,0%	5.3%	0,0%	0,0%
Above 70%	*13.5%*	9%**	*7.7%*	17.2%*	*6.7%*	0,0%	0,0%	0,0%	0,0%	0,0%
n	74	67	38	57	45	72	68	38	58	45

Note: ** statistically significant on 0.05 level, * statistically significant on 0.1 level, difference to the value in italics. Source: authors based on the research results.

Lean management techniques are therefore particularly important for organisations with more employees. A network approach based on an ecosystem of micro enterprises working together on a long-term basis could be a solution. This would make it possible to carry out larger projects, taking advantage of the cost-effectiveness of micro-enterprises. Such a strategy may be particularly feasible in the largely location-independent computer programming industry.

The main losses identified by respondents were delays and waiting, unnecessary meetings and switching between tasks (Table 2). Delays result from, among other things, decision latency. The moderately important losses are over-quality - unnecessarily added features and too much work in progress, unnecessarily implemented processes and wastage due to the use of outdated technology, and software bugs/defects. The least important types of wastage are delegation of work to other teams or individuals and unnecessary movement

of people, work and knowledge that hinders the smooth execution of tasks. Switching between tasks and losses due to the use of outdated technology were more common in CEE. In WE, significantly more respondents mentioned wastage in the form of too much work in progress and untapped employee creativity. Switching between tasks was most common in small enterprises, while losses due to software bugs/defects were most common in medium-sized and large companies. Unused employee creativity and losses due to the use of outdated technology were more common in small enterprises than in micro, medium and large enterprises.

Table 2. Types of waste the most common in company's software development projects.

Category	Europe (CEE+ WE)	Location		Numer of employees		
		CEE	WE	50 and more	10-49	Less than 9
Unnecessary meetings	51.4%	*55.4%*	47.1%*	56.4%	50%	48.9%
Switching between tasks	47.2%	*58.1%*	35.3%**	53.8%**	53.5%**	*33.3%*
Losses due to software bugs/defects	15.5%	18.9%	11.8%	23.1%**	13.8%	*8.9%*
Unnecessary movement of people, work, knowledge hindering smooth execution of tasks	9.9%	10.8%	8.8%	5.1%	12.1%	8.9%
Over-quality - unnecessarily added features	23.9%	25.7%	22.1%	28.2%	19,0%	26.7%
Too much work in progress	25.4%	*17.6%*	33.8%**	18%*	24.1%	*33.3%*
Untapped creativity of employees	13.4%	*6.8%*	20.6%**	10.3%	19%**	*6.7%*
Delegation of work to other teams or individuals	7.8%	9.5%	5.9%	7.7%	8.6%	6.7%
Unnecessarily implemented processes	19%	17.6%	20.6%	18,0%	19%	20%
Losses that may occur due to the use of outdated technology	19,0%	*24.3%*	13.2%**	18%	27.6%***	*6.7%*
Delays and waiting	52.8%	56.8%	48.5%	46.2%	51.7%	57.8%
n	142	74	68	38	58	45

Note: *** statistically significant on 0.01 level, ** statistically significant on 0.05 level, * statistically significant on 0.1 level, difference to the value in italics. Source: authors based on the research results.

The main causes of wastage in software development projects are overly ambitious customer requirements and running multiple projects at the same time (Table 3). Other important causes are an overly ambitious schedule, poorly diagnosed customer requirements, too small a project budget, poor communication within the project team, and a lack of effective monitoring and control. The least important causes are excessive responsibilities of the project manager, poor distribution of tasks among project team members, inadequate staff skills and the use of outdated technology. A specific problem for respondents from CEE is the lack of effective monitoring and control, and for WE companies the excessive responsibilities of the project manager. For medium and large companies, overly ambitious customer requirements and schedules, poorly diagnosed customer requirements and inadequate staff skills are particularly problematic. Larger organisations have more customers and therefore more problems in communicating with them. Small organisations are more likely than other types to report switching between tasks, poor communication within the project team, poorly diagnosed customer requirements and inadequate staff skills. It seems that more staff means a more diverse workforce in terms of skills. In addition, managers may be more focused on management issues and notice deficiencies in teams and employees. A higher number of projects in small enterprises trying to scale up leads to more frequent changes between tasks. A specific problem for micro enterprises is the excessive tasks of the project manager. It is likely that in micro enterprises the project manager has multiple roles in software development.

Other problems identified by companies were mainly related to customer issues, such as "problems caused by missing or changed customer data, insufficient customer involvement despite requests, immaturity and over-ambitiousness of customer representatives, inability of customer representatives to make decisions, excessive time spent on customer-side approvals,

inability of customer representatives to formulate project objectives, lack of customer-side knowledge of their own processes". Imprecise business requirements and too many changes to specifications were also mentioned. Sub-optimal team size is also a cause. As one respondent noted, "Sometimes a larger team is organised to get the project done faster, which is not effective because it takes much more time to organise the work than in a smaller team". Micromanagement and too little automation were also identified as causes of waste.

Table 3. The main causes of waste in the implementation of software development projects in the company.

Category	Europe (CEE+ WE)	Location		Number of employees		
		CEE	WE	50 and more	10-49	Less than 9
Poor communication within the project team	31.7%	33.8%	29.4%	28.2%	37.9%**	*24.4%*
Poorly diagnosed customer requirements	39.4%	*44.6%*	33.8%*	51.3%***	41.4%**	24.4%
Overly ambitious schedule	40.1%	39.2%	41.2%	53.9%**	*36.2%*	*33.3%*
Execution of several projects simultaneously	47.9%	44.6%	51.5%	46.2%	56.7%**	*37.8%*
Poor distribution of tasks among project team members	12%	10.8%	13.2%	12.8%	10.3%	11.1%
Lack of effective monitoring and control	30.3%	*37.8%*	22.1%**	30.8%	34.5%*	*22.2%*
Overly ambitious customer requirements	54.9%	55.4%	54.4%	69.2%**	*46.6%*	53.3%
Excessive tasks of the project manager	10.6%	*2.7%*	19.1%***	5.1%***	5.2%***	*20%*
Too small project budget	36.6%	*41.9%*	30.9%*	30.8%	39.7%	35.6%
Insufficient staff qualifications	15.5%	16.2%	14.7%	28.2%***	15.5%**	*4.4%*
Use of outdated technology	12.7%	10.8%	14.7%	15.4%	12.1%	8.9%
n	142	74	68	38	58	45

Note: *** statistically significant on 0.01 level, ** statistically significant on 0.05 level, * statistically significant on 0.1 level, difference to the value in italics. Source: authors based on the research results.

The most popular techniques for reducing wastage are agile project management (Kanban, Scrum, Product Increment), proper planning of time and tasks (realistic schedules and deadlines), better communication (e.g. inter-team meetings, frequent communication, especially in problem situations), and supervision by experienced people and the use of project management software (Table 4). Better communication was cited almost as often as agile methods, so it is likely that the latter are usually used to achieve this. Supervision and the use of software are especially important in CEE countries.

Table 4. Techniques used to reduce wastage in software development projects.

Category	Europe (CEE+ WE)	Location		Numer of employees		
		CEE	WE	50 and more	10-49	Less than 9
Agile project management	66.7%	*71.6%*	60.9%*	82.1%***	66.1%	*53.5%*
Proper planning of time and tasks	62.3%	62.2%	62.5%	61.5%	60.7%	65.1%
Better communication	62.3%	63.5%	60.9%	66.7%	62.5%	55.8%
Supervision by experienced employees	54.4%	*68.9%*	37.5%***	69.2%***	58.9%***	*34.9%*
Monitoring of the current workload of employees	29.7%	*40.5%*	17.2%***	35.9%*	28.6%	*23.3%*
Use of project management software	43.5%	*52.7%*	32.8%***	53.8%**	44.6%	*32.6%*
Better recruitment of suitable employees	29%	32.4%	25%	33.3%*	32.1%	*18.6%*
Pair programming	11.6%	*8.1%*	15.6%*	10.3%	8.9%	16.3%
Proactive detection of wastage	15.2%	14.9%	15.6%	25.6%**	*8.9%*	14%
n	138	74	64	38	56	43

Note: *** statistically significant on 0.01 level, ** statistically significant on 0.05 level, * statistically significant on 0.1 level, difference to the value in italics. Source: authors based on the research results.

Monitoring the current workload of employees is also a more popular technique among respondents from CEE. The least used techniques are pair programming (especially in CEE) and proactive wastage detection. Agile project management and supervision by experienced staff, as well as project management software, are used mainly by medium and large companies. They also stand out in proactive wastage detection. Other techniques mentioned by respondents include: training in requirements management, communication, looking after well-being/mental comfort and strategic advice to clients, and the right choice of project management methodology. However, one respondent noted that "despite popular opinion, agile methods are very often not suitable for implementing software projects. They only shift all risks onto the client's side, but do not help reduce losses or meet targets."

Table 5. Contingency table types of losses and causes of them.

Category	Losses due to bugs/ defects	Unneces-sary move-ment	Too much WIP	Untap-ped creati-vity	Delegatio n of work	Losses due to outdated techno-logy	Delays and waiting	n
Poor communication	10	10 (11.3***)	10	10 (4.4**)	2	13 (4.2**)	29 (3.6*)	45
Poorly diagnosed requirements	13 (4.21**)	6	15	10	4	12	29	56
Switching between tasks	9	8	21	10	9 (5.5**)	19 (6.8***)	36	68
Poor distribution of tasks	5 (2.9*)	4 (4.1**)	5	2	1	4	8	17
Lack of effective monitoring and control	16	5	8	7	5	11	20	43
Excessive tasks of the project manager	(22.2***)	2	7 (4,0**)	5 (5,8**)	2	1	7	15
Too small project budget	3	2	7 (4.0**)	5 (5.8**)	2	1	7	15
Insufficient staff qualifications	9	6	11	6	6	14 (3.3*)	34 (5.2**)	52
Use of outdated technology	10 (17.8***)	2	6	4	2	6	13	22
Unnecessary movement	.	.	.	.	.	.	11 (4.13**)	14
Unnecessarily implemented processes	.	.	.	.	.	.	9 (5.08**)	27
n	22	14	36	19	11	27	75	

Note: in brackets value of Chi-square Pearson test, ***- statistically significant on 0.01 level, ** statistically significant on 0.05 level, * statistically significant on 0.1 level. Source: authors based on the research results.

Among the types of losses, only a strong co-occurrence of delays and waiting and losses of unnecessary movement and unnecessary processes was observed (Table 5). Poor communication was an important cause of losses in the form of unnecessary movement, untapped creativity of employees, losses due to outdated technology, and delays and waiting (Table 5). Only this cause was associated with four types of wastage. Running several projects at the same time is a characteristic cause of losses in the form of delegation of work and wastage due to outdated technology. Excessive tasks of the project manager are a common cause of too much work in progress, and untapped creativity of employees. Poor distribution of tasks among project team members co-occurred abnormally often with unnecessary movement, and losess due to defects. Too small a project budget resulted in losses due to outdated technology, and delays. This means that the limited budget makes it difficult to modernise and employ more or more skilled staff, which could reduce delays. Software defects were mainly caused by lack of monitoring and control, poorly diagnosed customer requirements, and insufficient skills. The use of obsolete technology was a clear and expected cause of losses due to outdated technology.

Logistic regression showed that for every unit increase in the number of causes of wastage reported by a respondent from a computer programming company, the likelihood of having a strong position in the country's industry as well as good and very good development prospects decreased (Table 6). An increase of one in the number of wastage reduction techniques

implemented increases the probability of predicting good or very good development prospects for enterprises by 1.2. Medium and large companies are 2.9 times more likely to have a strong position in the industry than small and micro companies. A unit increase in the number of causes of wastage in software development projects in organisations increased the odds of having more than 50% of projects over time, scope or budget by 1.3 (compared to those having lower levels of problematic projects). Having foreign clients reduced the odds of having more than 50% of problematic projects. Probably exporting IT services with high revenues is likely to increase careful management of projects.

Table 6. Estimations of logistic regressions.

Explained variables	**Very good and good prospects for future**	**Over 50% projects significantly exceeding the budget, schedule or scope**		**Strong position in the industry in the country**	
Explaining variables	**Odds ratio(z-statistic)**				
Less than 10% wastage	.	0.2(-3.3***)	.	.	.
Foreign customers	.	0.4(-1.9*)	0.5(-1.8*)	.	.
Number of technics	1.2 (1.7*)	.	0.9(-0.95)	1(-0.07)	.
Number of causes	0.8(-2.6**)	.	1.3(2.2**)	0.7(-3.1***)	0.7(-3.1***)
CEE countries	.	.	.	1.2(0.5)	.
Over 49 employees	.	.	.	2.7(2.3**)	2.9(2.5**)
Constant	.	1.3(0.6)	0.45(-1.4)	1.4 (0.65)	1.4(0.8)
Pseudo R2	0.05	0.1	0.06	0.08	0.08
n	139	139	138	139	142

Note: in brackets value of z statistic Chi-square Pearson test, ***- statistically significant on 0,01 level, ** statistically significant on 0,05 level, * statistically significant on 0,1 level. Source: authors based on research.

Having a low level of wastage in projects (less than 10%) reduced the odds of having a high proportion (over 50%) of projects over scope, over budget or over schedule. This means that eliminating wastage and its causes reduces project inefficiency and is a predictor of having a strong position in the industry. Location in CEE or WE was not important for declaring a strong competitive position.

5. Conlusions

5.1. Conclusions for researchers

Sycz's [4] survey of IT experts found that unnecessary meetings were an additional type of loss characteristic of software development projects compared to the Toyota Production System. The current study of European computer programming companies confirms this. Moreover, our study analysed the opinions of representatives of 142 companies, mainly CEOs, directors and managers in different European countries. The research sample allowed comparisons to be made between CEE and WE countries and between companies of different sizes. As a result, it provides a more comprehensive analysis of the problem of waste in software development and offers a broader perspective of the entire enterprise than that of experts. However, the limitation of the research is the low response rate, which is characteristic for internet-based interviews. This does not allow the conclusions to be generalised to all European CP companies. They only reflect the opinions and situation of the responding companies. Future surveys could use face-to-face interviews or telephone/online meetings, which would increase the response rate. However, this would require a large budget. Further research on wastage in IT projects could also include a category of administrative burden as in the research of [6].

Our study showed that the most important types of losses in software development are delays and waiting, unnecessary meetings, and switching between tasks. Higher levels of wastage are characteristic of larger organisations. The percentage of respondents from WE who reported the lowest proportion of wasteful activities was higher than in CEE. It seems that they are more advanced in eliminating wastage. However, the highest proportion of problematic projects occur in small companies. Wastage is most often caused by over-ambitious customer requirements, especially in medium and large companies. They have more customers and customer relationship management seems to be crucial. Switching between tasks is the most visible in small companies. They are trying to scale up and take on more projects, but still have limited staff and budgets. For micro companies, the most common cause of wastage is the

excessive tasks of the project manager. In software development projects for micro-enterprises, they are likely to have to perform very different functions, accompanied by a greater administrative burden. A specific problem for respondents from CEE is the lack of effective monitoring and control, and for WE companies excessive tasks of the project manager.

In testing hypothesis H1, we found that losses caused by delays and waiting times are often accompanied by unnecessary processes, and unnecessary movement of people, work and knowledge, which hinder the smooth execution of tasks. This means that microproject techniques, according to flow modernisation with less complex work, should also reduce delays and waiting times. The analysis to verify hypothesis H2 showed that, in particular, poor communication within the project team leads to multiple types of losses. It causes unnecessary movement, untapped creativity of people, losses due to outdated technology, and delays and waiting. Stable, self-organising teams can reduce poor communication. Software bugs in particular have multiple causes. They are caused by a lack of monitoring and control, poorly diagnosed customer requirements and inadequate skills, but also by a poor distribution of tasks. Logistic regressions to validate hypothesis H3 have shown that the elimination of waste correlates with a company's better competitive position and development prospects. Therefore, the implementation of wastage elimination methods should improve the situation of companies.

5.2. Conclusions for practicioners

Lean IT and Le-agile are becoming increasingly popular in software development project management because they are more pull-oriented methods - producing only what the customer wants - than Agile. They overcome administrative burdens with less managerial effort due to higher levels of self-organisation and self-control of teams. Eliminating wastage is crucial. Techniques used by CP companies in Europe to reduce wastage include, in particular, agile project management and proper planning. Supervision and the use of project management software are particularly important in CEE countries. Monitoring the current workload of employees is also a more popular technique among CEE respondents. Medium and large companies are more advanced in proactively identifying wastage.

Better communication and meeting management seems to be crucial for software development efficiency. Due to the problems caused by poor communication, Le-agile seems to be an appropriate technique that would both improve communication and optimise costs by eliminating wastage. It requires soft skills that may be problematic for some introverted IT professionals. So perhaps the ability to codify communication through short notes in project management software could be a solution. As wastage is often due to poor customer relations, perhaps some training should be offered by IT companies to client companies on how to manage IT projects from the customer side. With well-prepared staff, client companies would be better able to articulate their requirements and better absorb the software developed. Eliminating wastage improves the development prospects of companies. Therefore, the popularisation of Lean IT and wastage elimination methods, including through the academic training of IT professionals, should increase the efficiency of software development projects.

References

1. Johnson, J., Mulder, H.: Endless Modernization: How Infinite Flow Keeps Software Fresh, The Standish Group International, Inc. (2020)
2. Schwaber, K.: Agile Project Management with Scrum, Microsoft Press (2004)
3. Grobelna, K., Trzcieliński, S.: Agile Software Development Methodologies vs. Contemporary Management Concepts (in Polish). Przedsiębiorczość i Zarządzanie, Wydawnictwo SAN, XVIII (4) I (2017)
4. Sycz, P.: Methodology for reducing wastage in IT projects (in Polish). Doctoral dissertation, University of Gdansk, Sopot (2023)
5. Jorge, G.F.: Continuous Improvement and Waste Reduction for Leaner IT Organizations – an Introduction to Lean IT Project Management Development – Practice and Perspectives. Sixth International Scientific Conference on Project Management in the Baltic Countries April 27-28, Riga (2017)
6. Sedano, T., Cécile, P.R.: Software Development Waste, International Conference on

Software Engineering, ICSE (2017)
7. Dennehy, D., Conboy, K.: Identifying Challenges and a Research Agenda for Flow in Software. Project Management Project Management Journal. 49(6), 1–16 (2018)
8. Sallin, M., Kropp, M., Anslow, C., Biddle, R.: Waste Self-reporting for Software Development Productivity Improvement. In: Stettina, C.J., Garbajosa, J., Kruchten, P. (eds) Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing 475, Springer. Cham (2023)
9. Power, K., Conboy, K.: Impediments to Flow: Rethinking the Lean Concept of 'Waste' in Modern Software Development. Lecture Notes in Business Information Processing. 179 LNBIP, pp. 203–217 (2014)
10. Ikonen, M., Kettunen, P., Oza, N., Abrahamsson, P.: Exploring the Sources Of Waste in Kanban Software Development Projects. Proceedings - 36th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA, pp. 376–381 (2010)
11. Aldaeej, A., Nguyen-Duc, A., Gupta, V.: A Lean Approach of Managing Technical Debt in Agile Software Projects – A Proposal and Empirical Evaluation. In: Stettina, C.J., Garbajosa, J., Kruchten, P. (eds) Agile Processes in Software Engineering and Extreme Programming. Lecture Notes in Business Information Processing. 475, Springer. Cham (2023)
12. Monserrat, M., Mas A., Mesquida Calafat, A.L., Clarke, P.: Applying Lean to Improve Software Project Management Education. IEEE Transactions on Engineering Management. 71, pp. 7496-7510 (2024)
13. Weflen E., MacKenzie C.A., Rivero I.R.: An Influence Diagram Approach to Automating Lead Time Estimation in Agile Kanban Project Management. Expert Systems with Applications. 187 (2022)
14. Gunawan, F., Budiardjo, E. K.: A Quest of Software Process Improvements in DevOps and Kanban: A Case Study in Small Software Company. ACM International Conference Proceeding Series, pp. 39–45 (2021)
15. Cvetkovic, N., Morača, S., Jovanović, M., Medojević, M., Lalić, B.: Enhancing the Agility and Performances of a Project with Lean Manufacturing Practices. Annals of DAAAM and Proceedings of the International DAAAM Symposium, 661–670 (2017)
16. Estevam, A., Dennehy, D., Conboy, K.: Using Flow Tools to Enact Control in Software Development Projects: A Cross-case Analysis. Inf Syst Front 24, 287–304 (2022)
17. O'Connor, M., Conboy, K., Dennehy, D., Carroll, N.: Temporal Complexity in Information Systems Development Flow: Challenges and Recommendations, Communications of the Association for Information Systems. 54 (2024)
18. Guedes, J.F.: A Lean Mind-set on the Information Technologies Sector: Targeting and Addressing Waste for an Increased Performance. In: Mejia, J., Muñoz, M., Rocha, Á., Quiñonez, Y., Calvo-Manzano, J. (eds) Trends and Applications in Software Engineering. CIMPS 2017. Advances in Intelligent Systems and Computing. 688, Springer. Cham (2018)
19. Kobus, J., Westner M., Strahringer S., Strode D.: Enabling Digitization by Implementing Lean IT: lessons learned. TQM Journal. 30, (2018)
20. Womack, J.P., Jones, D.T.: Lean Thinking: Banish Waste and Create Wealth in Your Corporation, Simon & Schuster, New York, NY (1996)
21. Middleton, P., Joyce, D.: Lean Software Management: BBC Worldwide Case Study. IEEE Transactions on Engineering Management. 59(1), pp. 20–32 (2012)
22. Szczepańska, K.: Total Quality Management. Past and Present (in Polish), Publisshing House of Warsaw Univrsity of Technology, Warsaw (2010)
23. Vanzant Stern, T.: Lean and Agile Project Management How to Make Any Project Better, Faster, and More Cost Effective, CRC Press (2017)
24. Mehta, M., Anderson, D., Raffo D.: Providing Value to Customers in Software Development through Lean Principles. 13(1). Special Issue:Advances in Software Process Improvement, pp. 101-109. January/February (2008)
25. Szymczak, W.: Fundamentals of Statistics for Psychologists (in Polish), Difin (2010)
26. Greene, W.H.: Econometric Analysis, Prentice Hall (2003)