Thesis Summary

Extending Software Project Scheduling Problems to Investigate Group Selection Mechanisms

Author:
Péter Harta

Supervisor:
Prof. Dr. habil. Zsolt Tibor Kosztyán

University of Pannonia

Doctoral School in Management Sciences and Business Administration
Department of Quantitative Methods

February 26, 2025

1 Introduction

Today's, software undeniably plays a crucial role, thanks to the widespread adoption of automation (the Fourth Industrial Revolution) (Wankhede and Vinodh 2021). In software development projects carried out by collaborative teams, particularly those following an agile approach, autonomous (Hoda et al. 2012, Kanski et al. 2023), cross-functional (Gutierrez et al. 2018, Meier and Kock 2023), and diverse teams (Galinsky et al. 2015, Albusays et al. 2021, Guimera et al. 2005) are currently preferred.

In the literature, the characteristics of software development project team members have already been considered in project planning within the framework of Software Project Scheduling Problems (SPSP) (Alba and Chicano 2007, Vega-Velázquez et al. 2018, Rezende et al. 2019), where the optimal assignment of software engineers to project tasks was modeled, taking into account their skills and the resource requirements of the project. Moreover, in recent years, researchers have recognized that, in addition to the qualifications of project team members, the synergy among them also has a significant impact on the success of software projects (Kosztyán et al. 2022, Dwertmann et al. 2016, Liemhetcharat and Veloso 2012), which is correlated with the employees' soft skills (Pieterse et al. 2018).

Despite the widespread use of software project scheduling problems, nearly 50% of such projects are failed or challenged (VersionOne 2024, Group 2021), due to inadequate team structures arising from collaboration issues (Aryanee et al. 2020, Gilal et al. 2016), non-complementary team roles (Vishnubhotla et al. 2018, Bell et al. 2018), or the failure to account for the dynamic skills (Schulze and Brusoni 2022).

In software project scheduling, team heterogeneity stems from skill differences caused by cross-functionality or diverse team roles, which can lead to both conflict and collaboration (Lee et al. 2015, Zainal et al. 2020). The heterogeneous network of such teams can already be modeled by leveraging the individual attributes of team members and the relationships between them. This is described in the scheduling model by Kosztyán et al. (2022). However, the practical applicability of this method could be significantly improved if synergistic effects could be quantified and soft skills, which better describe team dynamics, could also be considered. A potential solution is to incorporate various team roles and their characteristics into scheduling. Despite the fact that agile and hybrid approaches have been central philosophies in software development projects for over 20 years, the modeling of cross-functional teams as networks composed of heterogeneous roles has not yet been integrated into software project scheduling.

With such a scheduling method, it would be possible to analyze the autonomous organization of a heterogeneous team, as well as the dynamic changes caused by turnover, in terms of team capabilities and the synergistic effects among different team members.

The objective of this dissertation is twofold. First, it aims to develop and propose a novel project scheduling method that explicitly accounts for heterogeneous team networks. The proposed method will show how to model a software project, where

- soft skills and hard skills are separated,
- synergies and heterogeneous behavioral types / team roles of employees are considered, and
- flexible dependencies are managed.

Second, it seeks to evaluate the effectiveness of autonomous teams and central team roles in project scheduling through this new approach.

Throughout the dissertation, I interpret the project as an external project. In this context, two main actors of the projects can be distinguished: the project owner organization that initiated the project (hereinafter referred to as project owner or customer), and the project-based organization who perform the project. I interpreted my research from the point of view of the organization performing the project, i.e. I deal with the investigation of the performance of the ordered project.

2 Research questions

Considering the significance of the topic and the previously stated objectives, this thesis seeks to address the following research questions:

RQ$_1$ How can a software project scheduling method be enhanced to incorporate the uniqueness of different team roles and behavioral types, while also considering their interactions within a heterogeneous network, shaped by diverse skill sets and synergy effects, in both structured and flexible environments?

RQ$_2$ How do central team roles as the central unit of a heterogeneous network influence the success of software projects through their integration into scheduling strategies?

RQ$_3$ How does autonomously selected team as a heterogeneous network affect the success of software projects through their scheduling?

3 Literature review

Evolution of software project management

The conceptual interpretation of projects has undergone changes over the past 40 years (Wawak and Woźniak 2020, Shenhar and Dvir 2007, Görög 2003). Initially, projects as processes were effectively described using the iron triangle (De Wit 1988) which is defined by the traditional project management approach, which framed the efficient completion of project outcomes in terms of quality, duration, and cost (Wysocki 2019). In the case of software development projects, the project outcome is specified, that is, the deliverable is a new software product created by at least two individuals (Wysocki 2010). Under this approach, software project management traditionally focused on project time, cost, and resource planning, along with project control, to deliver a new software solution. This framework was established by traditional project management methodologies (Wysocki 2019), which described the project life cycle using the waterfall model.

However, in the early 1990s, two new perspectives emerged: one that viewed projects as temporary organizations (Lundin and Söderholm 1995, Söderlund 2004), and another that considered projects as building blocks of strategy (Cleland 1994). The "project as a temporary organization" perspective assumes that the executing organization functions as a separate entity during the project's duration, incorporating aspects of team dynamics and change management. Meanwhile, the "project as a strategic building block" perspective suggests that projects are initiated to provide a framework for favorable changes defined within an organization's strategy. Both perspectives introduced shifts in project life cycle models and in how project success is evaluated. Life cycle theories have shifted towards iterative and incremental approaches (Wysocki 2010), while new success criteria, such as stakeholder and project team satisfaction, have been added to the traditional project management triangle (Görög 2003, Lundin and Söderholm 1995, Wawak and Woźniak 2020). This evolution paved the way for the agile project management approach, which originally emerged in software development (Beck et al. 2001). Agile methodologies introduced iterative and incremental project execution frameworks such as SCRUM (Schwaber and Beedle 2001, Meckenstock 2024), which significantly involve project stakeholders in the execution process. Although this approach was already a fundamental aspect of project management before the formal introduction of agility as a concept, the Agile Manifesto (Beck et al. 2001) served as a milestone that formally outlined the principles of this mindset.

Due to the popularity of agile methodologies and the effectiveness demonstrated in industry reports (VersionOne 2024, Group 2021), agile approaches have become widely adopted across various sectors (Salo and Abrahamsson 2008, Gupta et al. 2022). However, this widespread adoption also led organizations unprepared or unsuited for agility to attempt agile transformations, resulting in a high failure rate for agile projects (Group 2015, 2021). Consequently, hybrid approaches have gained traction (Reiff and Schlegel 2022) which are combining traditional and agile methodologies, and allowing for a gradual and rational implementation of agility.

Both agile and hybrid approaches place software development teams at the center of project execution (Dybå et al. 2014). This central role is so significant that both methodologies emphasize the importance of team dynamics (Meckenstock 2024). One of the core pillars of agility is autonomous teams, which self-organize and take on tasks without external intervention (Gupta et al. 2022, Schwaber and Beedle 2001). However, these agile and hybrid approaches are only effective at the team level when applied within organizations that have stable human resources—meaning companies with low employee turnover. In the software industry, however, turnover is a significant factor. Since turnover primarily affects initial project scheduling, it is essential to review software project scheduling methodologies accordingly.

Software project scheduling

The scheduling of software development projects is essentially an NP-hard resource allocation problem (Xiao et al. 2013), where the goal is to assign the right person to the right task while considering objective functions and constraints (Alba and Chicano 2007). Initially, when the focus was solely on single-objective optimization, Resource-Constrained Project Scheduling Problems (RCPSP) effectively described this process (Blazewicz et al. 1983, Hartmann and Briskorn 2022). In the context of software development, this problem becomes particularly interesting when team members' skills are considered as resources, whether it involves multiple skills, called Multi-Skilled Resource-Constrained Project Scheduling Problems (MS-RCPSP) (Hegazy et al. 2000, Myszkowski et al. 2017), multiple distinct resources, called Multimode Resource-Constrained Project Scheduling Problems (MRCPSP) (Coelho and Vanhoucke 2011), or a combination of both, called Multi-Skilled Multimode Resource-Constrained Project Scheduling Problems (MS-MRCPSP) (Maghsoudlou et al. 2016). All these problems are essentially resource allocation problems, assuming that each activity has a predefined duration (Myszkowski et al. 2017). However, in software development, task durations are not fixed but rather depend on the required skill levels for the task and the competencies of the assigned individuals. This challenge is addressed by the class of Software Project Scheduling Problems (SPSP), which not only considers the factors mentioned above but also involves multi-objective optimization. The goal of SPSP is to achieve efficient resource allocation while minimizing both project duration and cost (Alba and Chicano 2007, Vega-Velázquez et al. 2018, Rezende et al. 2019). Various factors have been incorporated into the class of SPSP (Vega-Velázquez et al. 2018, Rezende et al. 2019), such as flexible tasks (Zapotecas-Martínez et al. 2020, Kosztyán et al. 2022), multi-skills (Li et al. 2023), level of skills (García-Nájera and del Carmen Gómez-Fuentes 2014), synergetic effects among team members (Kosztyán et al. 2022), and learning effect (Cheng et al. 2019). Beyond that, numerous solution algorithms have been developed to solve this NP-hard problem, including Genetic Algorithms (Deb et al. 2002), Ant Colony Optimization (Xiao et al. 2013), and Grey Wolf Optimization (Alabajee et al. 2021). Among these, the most relevant to this dissertation is the synergy-based Software Project Scheduling Problem (SSPSP) presented by (Kosztyán et al. 2022), which accounts for multiple skills, level of skills, flexible project planning, and the team's synergy. Despite the effectiveness of SSPSP in optimizing team resource allocation, no existing method considers heterogeneous team structures, heterogeneous roles within the team, or the role of the network of these heterogeneous roles in scheduling. Understanding these aspects is crucial for analyzing factors such as the efficiency of autonomous team organization and the impact of the team's central unit on the success of projects.

Individual diversity and team network

A heterogeneous team is a synergistic organization of people with different abilities and behaviors working towards a common goal (Groysberg et al. 2011, Bell and Outland 2017). Heterogeneity can be interpreted in multiple ways (Jehn et al. 1999). Differences between people can be information-based, primarily arising from their skills. In this view, this could manifest in the level or type of skills. Social differences, which include personality and behavioral differences, are equally important. Finally, there can be attitude differences depending on how much individuals align with the team's goals, and the attitudes they exhibit during teamwork (Jehn et al. 1999).

The organization of the team and the organization of its network is also a process, where individual behaviors transform into collective team roles. This organization is described by the Tuckman model (Tuckman 1965), which highlights that even in autonomous team organization, certain steps are needed for individuals to form a team and a stable network. The Tuckman model essentially consists of five main phases: (1) forming, (2) storming, (3) norming, (4) performing, (5) adjourning. If we map this to team organization, autonomous team organization corresponds to the forming phase, where individuals are most influenced by their behavior types. In this phase, the properties of the network are still shaped by flexible relationships, and team roles have not yet formed. In contrast, in a team that has already formed, where a central unit has developed, the norming phase will be characteristic. Here, team roles start to emerge.

In software project scheduling, skills are often used for scheduling, ignoring the different levels of team formation. These skills can generally be divided into two groups: technical or hard skills and social or soft skills (Balcar 2016). In software development, the necessary hard skills are typically project-dependent but can often be defined based on the amount of code written and tested, or the English language documentation produced (Matturro 2013, Hidayati et al. 2020). Soft skills are much harder to quantify, but the most important soft skills in a software development team are defined in the literature as communication skills, leadership skills, teamwork attitude, problem-solving skills, analytical skills, and interpersonal skills (Hidayati et al. 2020, Matturro 2013, Borges and de Souza 2024).

Many researchers point out that heterogeneous teams are more successful than homogeneous teams (Peslak 2006, Phillips et al. 2009, Bear and Woolley 2011, Galinsky et al. 2015), while others have drawn the opposite conclusion (Towry 2003, Van Knippenberg et al. 2004, Hamilton et al. 2012, Waleed et al. 2021). It can therefore be assumed that in a heterogeneous network, both positive and negative relationships can exist between individuals.

Behavioral styles and roles formed within the team are no longer considered in the scheduling of software development projects in this manner. Autonomous team organization, being established to complete a task, does not have time to go through the team organization process. In this case, the individuals' behavior types primarily determine the synergistic effects between them. These behavior types can be described using the DISC behavioral type theory (Marston 1928, Lykourentzou et al. 2016, Scullard and Baum 2015). This theory divides people into four clusters based on two dimensions: introverted-extroverted (dI-E) dimension and task-oriented-relationship-oriented (dT-dR) dimension. Accordingly, four behavior types can be distinguished: (D) dominance (dE, dT), (I) influence (dE, dR), (S) steadiness (dI, dR), and (C) conscientiousness (dI, dT). Due to its simplicity and widespread use, the DISC behavioral type theory is commonly used for selecting small agile teams (Diekmann and König 2018, Reynierse et al. 2000, Lykourentzou et al. 2016). However, for describing already formed teams, the Belbin team roles theory based on divergent team roles is more applicable (Belbin 1981). This theory distinguishes 8 (later 9) team roles based on their functions within the team, such as: (CO) coordinator, (TW) team player, (RI) resource investigator, (IMP) implementer, (CF) completer, (SH) shaper, (PL) plant, (ME) monitor evaluator. Belbin grouped these roles into three categories: thinking-oriented (PL + ME), people-oriented (CO + TW + RI), and action-oriented (SH + CF + IMP) team roles.

To take both DISC behavioral types and Belbin team roles into account in software project scheduling, the factors causing their diversity and cooperation need to be determined. For DISC behavioral types, I used the work of Scullard and Baum (2015) and Marston (1928), and for Belbin team roles, I estimated the synergy potential network based on Belbin (1981) work. Based on these researches, positive synergy potential assumes that the combined work of two people is expected to add more value than their individual work. In contrast, a negative synergy potential value means that the two individuals are not effective together.

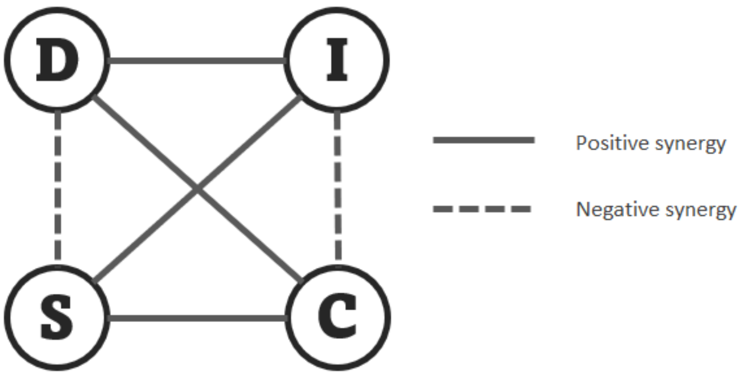


Figure 1: The estimated synergy network among DISC behavior types

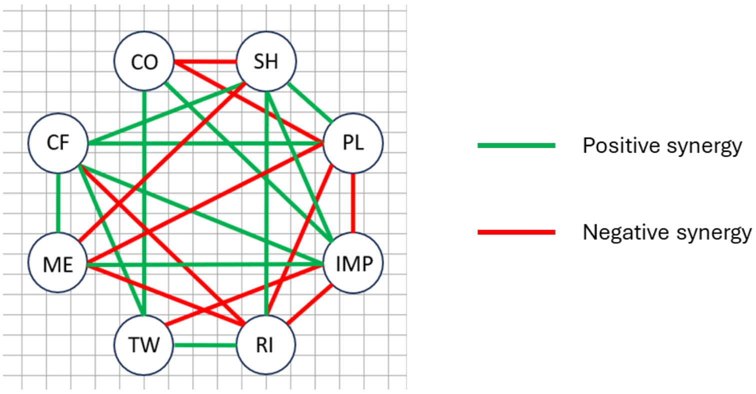


Figure 2: The estimated synergy network among Belbin's team roles

Referring back to the contradiction in the literature that heterogeneity can have both positive and negative effects on team performance, Figure 1 and Figure 2 offer a possible explanation for this contradiction. Indeed, the combination of skill-based and personality-based heterogeneity can influence the team's success (Lee et al. 2015, Zainal et al. 2020).

Thus, a heterogeneous team can be diverse in skills or personalities. In the above examples, selecting a team that is diverse in personality can be achieved and modeled using DISC behavioral types or Belbin's team roles. A team that is heterogeneous in skills is called a cross-functional team, which is another important aspect of agile and hybrid approaches (Kaliprasad 2005). Therefore, when selecting a team for an effective agile or hybrid environment, it is important to consider cross-functionality and autonomy (Beck et al. 2001, Hoda et al. 2010, Dingsøyr and Dybå 2012, Meslec and Curşeu 2015). If this team, or any other, later enters the norming phase, it will be important to maintain team stability, at which point the selection process will no longer be a strict selection but rather an effective allocation of team members to complete project tasks. Both autonomous cross-functional team selection and effective allocation of a team with a central unit can be realized if the SSPSP framework is integrated with personality type theories (in this case, behavioral type theory and team role theory). Autonomous team selection can be modeled using DISC behavioral types, as during the autonomous team formation, team roles have not yet developed, and the team's properties are influenced by the behavior of its members. In contrast, the influence of the central unit in an already formed team is well modeled using Belbin's team roles, as it is assumed that stable team roles have already been established. A strong argument for using Belbin's team roles is that, considering the positive relationships in their synergy network, the action-oriented group plays a central role (Figure 3.)

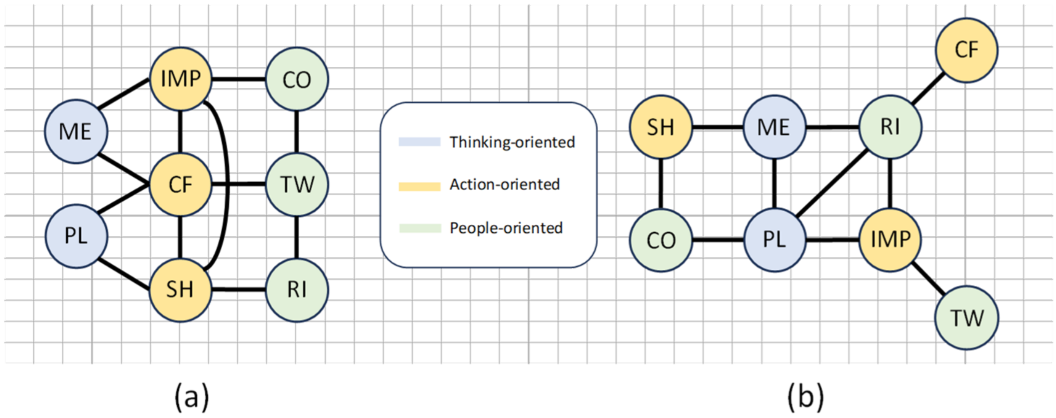


Figure 3:
Positive (a) and negative (b) synergistic relationships between Belbin's team roles. Team members of the thinking-oriented groups are marked with blue, the action-oriented group is marked with yellow and the people-oriented group members are marked with green.

3.1 Research assumptions

Drawing upon the background of the literature, I have formulated the following three research assumptions (RA_1, RA_2, RA_3) to align with the research questions (RQ_1, RQ_2, RQ_3)

RA$_1$ The SSPSP method can be expanded to incorporate Belbin team roles and DISC behavioral types by leveraging the synergies among these roles, as well as the soft and hard skills they represent, within a flexible software environment.

RA$_2$ The presence of central team roles in software projects positively impacts project success, thereby enhancing performance within the constraints and objective functions defined in the supplemented SSPSP.

RA$_3$ Autonomous teams positively impact the success of software projects, within the constraints and objective functions of the enhanced SSPSP, more effectively than teams with dedicated leaders.

4 Results

RQ$_1$ *How can a software project scheduling method be enhanced to incorporate the uniqueness of different team roles and behavioral types, while also considering their interactions within a heterogeneous network, shaped by diverse skill sets and synergy effects, in both structured and flexible environments?*

For the development of the method, I selected the SSPSP framework presented in Kosztyán et al. (2022) as the base method, as it was the closest to the desired approach. For data storage, I also used the synergy mapping model (SMM) introduced in Kosztyán et al. (2022).

Based on the objectives described in Section 1, soft skills were first taken into account. In this case, I decomposed the vector of skills ($[s]$) in the base model into hard skills ($[s_h]$) and soft skills ($[s_s]$) vectors. The values in the ($[s_h]$) vector are additive, whereas the values in the ($[s_s]$) vector are non-additive, meaning that, for example, taking their average does not make sense. Thus, the modified SMM matrix has a size of $m + n \times m + s_h + s_s + n + 1$, where m represents the number of people, n denotes the number of project tasks, s_h indicates the number of hard skills, and s_s specifies the number of soft skills.

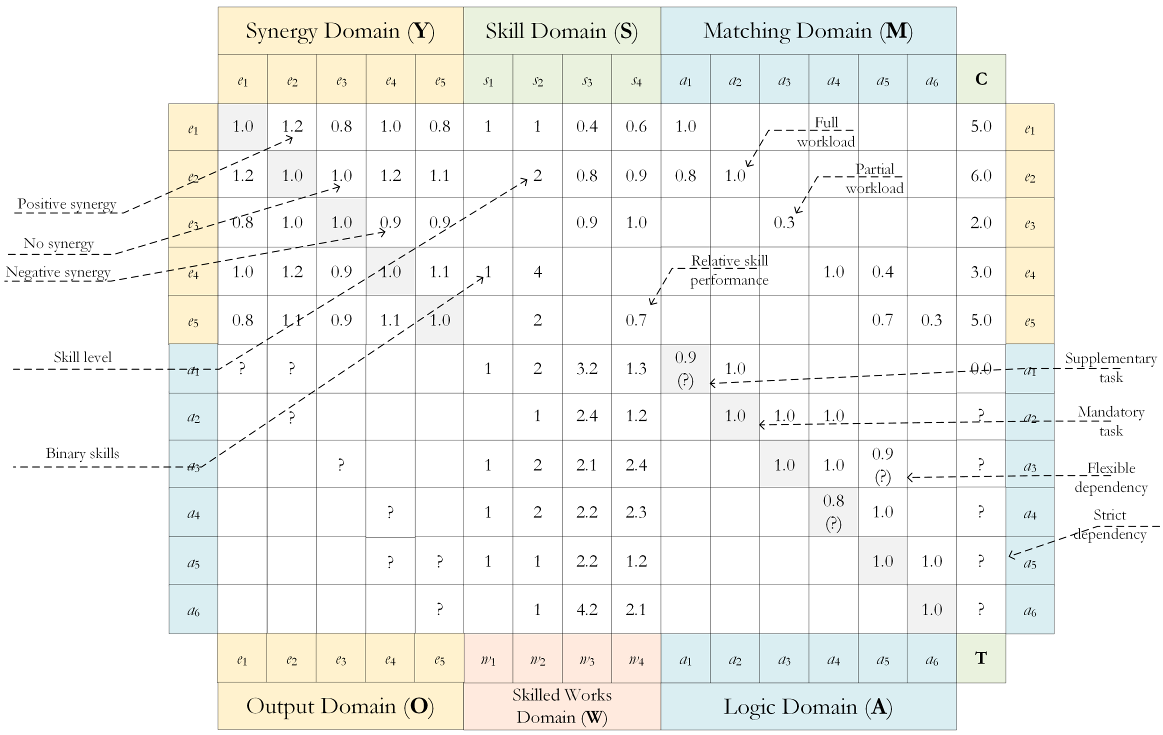


Figure 4: Modified SMM matrix

Secondly, the synergy potential values between behavior types and personality traits were stored in the model's "Synergy Domain (Y)" matrix, where only the upper triangular matrix is relevant due to duplication (the lower triangular matrix is the transpose of the upper one). Here, a value greater than 1 indicates a positive synergistic relationship, a value less than 1 suggests a negative synergy, and a value equal to 1 implies no synergistic effect between the two individuals.

In the model, additive skills are weighted by the geometric mean of the synergistic effects among the individuals performing a given task, thus considering the j-th skills::

$$S_j^{\varepsilon} := \overline{Y}_{\varepsilon} \cdot \sum_{i \in \varepsilon} [\mathbf{S}]_{ij} \tag{1}$$

where $\overline{Y}_{\varepsilon}$ a *geometric mean* of ε a subset of employees who work together on a given task of the project:

$$\overline{Y}_{\varepsilon} := \begin{cases} 1 & \text{if} \quad |\varepsilon| \leq 1 \\ \sqrt[\eta]{\prod\limits_{i,j \in \varepsilon,\ i<j} [\mathbf{Y}]_{i,j}} \qquad \text{where } \eta = \frac{|\varepsilon| \cdot (|\varepsilon|-1)}{2} & \text{if} \quad |\varepsilon| > 1 \end{cases} \tag{2}$$

As the third objective, I modeled the logical relationships between tasks in the "Logic Domain (A)" section of the model. According to this, tasks can be classified as mandatory (if $[\mathbf{A}]ii = 1$) or supplementary (if $[\mathbf{A}]ii < 1$). Additionally, dependencies can be distinguished as strict (if $[\mathbf{A}]ij = 1$) or flexible (if $[\mathbf{A}]ij < 1$).

Furthermore, the "Matching Domain (M)" section of the SMM model stores the maximum dedication of individual employees for a given task, which takes values in the interval $]0,1[$. Accordingly, employee i can contribute to task k with a workload of $[\mathbf{M}]_{ik}$. The required quantity of additive and non-additive skills for executing project tasks is summarized in the "Skilled Work Domain (W)". The optimal resource allocation is encoded in the "Output Domain (O)" matrix, which, similar to the Matching Domain, takes values in the range of $]0,1[$ depending on the proportion of an employee's available capacity assigned to a given task.

During optimal assignment, I performed multi-objective optimization considering total project time (TPT), total project cost (TPC), and total project score (TPS). Both TPT and TPC are functions of the maximum workforce required for task completion $(([\mathbf{W}]jk))$, the sum of the optimally allocated employees' skills $(([\mathbf{S}]ik * [\mathbf{O}]ji))$, and the synergistic effects among a subset of employees $(\overline{Y}\varepsilon_j)$, if considered. Additionally, TPC also depends on a salary vector $([C])$, which encodes the salaries of individual employees.

Since the problem is NP-hard, I used a hybrid genetic algorithm to solve it. This is a two-step solution method: in the first step, a genetic algorithm selects a set of possible solutions, and in the second step, the Nelder-Mead method is applied to find the best solution according to the objective functions.

The main difference compared to Kosztyán et al. (2022) is that the genetic algorithm's multi-chromosome encoding includes the possibility of selecting a specific selection method from a set containing multiple team selection methods. To ensure the genetic algorithm operates with sufficient accuracy, I tuned its hyper-parameters using the "Design of Experiments (DoE)" method. Accordingly, I used the genetic algorithm with the following hyper-parameter values:

Operators	**Parameters**
Population size	250
Badges of chromosomes	4
Turnament size	9
Elite count	0.05
Crossover fraction	0.82
Rate of feasible chromosomes in crossover	0.88
Probability of mutation of a gene in a chromosome	0.05
Maximal rate of migrated chromosomes	0.09
Tolerance value	$1E-8$
Maximal iteration	150

Table 1: Tuned hyperparameters of GA

The reliability of the method was also tested using the validated database from Myszkowski et al. (2019) and compared with other similar optimization methods. Table 2 shows a comparison of the results. These algorithms include (1) a simple duration-oriented heuristic algorithm, presented by Myszkowski et al. (2013); (2) a duration-oriented greedy algorithm; (3) the well-known metaheuristic ant colony algorithm (ACO); and (4) the modified hybrid ant colony algorithm (HAntCO), which can use priority rules against the simple ACO, where (2-3-4) are presented by Myszkowski et al. (2015). The original SSPSP algorithm was implemented by Kosztyán et al. (2022), and the modified, poposed SSPSP algorithm was used as the HGA algorithm. It should be noted that the projects in the iMopse database do not take into account synergistic effects, flexible task dependencies, skill performances, or allocation ratios.

Table 2: Comparision of existing methods in MS-RCPSP (n is the number of tasks, e is the number of employees, p is the number of precences, s is the number of skills, TPT is the total project time, TPC is the total project cost. ACO is the Ant Colony Optimization, HAntCO is a the modified (heuristic) Ant Colony Optimization, SSPSP is the algorithm for the synergy-based software scheduling problem.

				Heuristic		Greedy		ACO		HAntCO		original SSPSP		modified SSPSP	
n	e	p	s	TPT	TPC	TPT	TPC	TPT	TPC	TPT	TPC	TPT	TPC	TPT	TPC
100	10	26	15	37	126361	38	119336	32	124687	31	126216	35	124168	35	124154
100	10	27	9	38	44309	38	43438	34	44999	33	42199	37	43756	36	43750
100	10	47	9	41	142759	40	135161	36	143100	34	140865	38	140483	38	140489
100	10	48	15	36	135534	44	120664	33	133062	33	133495	37	130698	37	130693
100	10	64	9	39	113124	43	117993	35	110643	33	113774	38	113898	38	113892
100	10	65	15	40	152955	43	140782	35	150294	32	149185	38	148305	38	148313
100	20	22	15	25	117493	24	112135	20	120949	19	123642	22	118568	23	118556
100	20	23	9	32	53154	32	50279	32	52119	23	53358	30	52235	31	52242
100	20	46	15	28	138270	29	133739	25	138565	24	138568	27	137286	27	137294
100	20	47	9	21	129160	28	140626	21	124817	18	134312	22	132235	22	132247
100	20	65	15	32	110503	34	118569	27	109831	27	108991	30	111987	31	111974
100	20	65	9	25	127149	24	124291	23	130934	21	126659	24	127267	23	127261
100	5	20	9	57	40539	55	40958	50	41029	53	40811	55	40841	55	40849
100	5	22	15	63	119266	77	128354	60	119434	60	119158	65	121570	65	121555
100	5	46	15	75	202238	80	202607	67	204110	67	204730	72	203422	73	203437
100	5	48	9	72	193383	78	196893	62	191712	62	191888	69	193471	69	193487
100	5	64	15	71	141407	66	141882	62	144972	61	143956	65	143068	65	143073
100	5	64	9	71	102439	67	107014	61	102777	61	101297	65	103385	66	103399
200	10	128	15	71	180812	78	198378	62	178264	60	178375	68	183960	68	183969
200	10	135	9	216	105593	216	93426	216	99375	186	103561	209	100508	209	100492
200	10	50	15	66	189660	75	183673	63	191856	62	190956	67	189042	67	189045
200	10	50	9	66	251158	70	250732	65	250075	64	250850	67	250721	67	250717
200	10	84	9	70	224121	66	222976	69	226666	66	222655	69	224121	68	224110
200	10	85	15	65	304277	68	301357	61	306949	62	302064	65	303677	64	303682
200	20	145	15	36	275983	46	277097	36	278199	35	272504	39	275947	39	275956
200	20	150	9	183	92821	183	95667	186	91461	177	92567	183	93146	183	93143
200	20	54	15	37	295786	41	290656	39	299993	34	298822	38	296334	39	296330
200	20	55	9	37	230150	37	232766	38	231094	36	223879	38	229484	38	229486
200	20	97	15	49	290399	69	346527	42	280951	42	277860	51	298948	51	298935
200	20	97	9	35	273378	43	282379	37	275819	35	278797	38	277608	38	277596
200	40	130	9	112	101879	112	90907	112	94488	108	104965	112	98066	112	98079
200	40	133	15	24	276456	23	279170	27	281933	24	279073	25	279167	25	279178
200	40	45	15	31	260738	32	269623	25	248717	23	256687	29	258946	28	258942
200	40	45	9	22	270758	23	276416	26	273632	25	270428	25	272819	24	272824
200	40	90	9	24	290028	20	294909	26	287694	24	298340	24	292752	24	292758
200	40	91	15	19	249909	35	250843	25	257927	23	241492	26	250059	26	250049
			Mean:	54.61	176498.58	57.69	178117.31	51.94	176197.97	49.39	176027.19	53.92	176719.25	53.83	176719.44

According to the first research assumption (RA_1) *"The SSPSP method can be expanded to incorporate Belbin team roles and DISC behavioral types by leveraging the synergies among these roles, as well as the soft and hard skills they represent, within a flexible software environment."*. Based on Table 2, the new method is capable of finding the best solution with adequate accuracy in software project scheduling.

Since the accuracy and applicability of the new method could only be partially validated due to the specific characteristics of the validated iMopse database, it is necessary to further validate the method using empirical data.

RQ$_2$ *How do central team roles as the central unit of a heterogeneous network influence the success of software projects through their integration into scheduling strategies?*

RQ$_3$ *How does autonomously selected team as a heterogeneous network affect the success of software projects through their scheduling?*

To answer the research questions above, I have prepared a case study examining multiple cases (Yin 2009), which not only addresses the research questions but also empirically validates the correctness of the new model. The research was conducted at the R&D department of Continental Automotive Hungary Ltd. located in Veszprém, which employs nearly 500 people in the software development field. The summary of the research is provided in Table 3.

Table 3: Case Study Report

Timeline	Process activity	Extended explanation
October 2022 - March 2023	Formulate the theory	During this period, I developed the new SSPSP model and formulated the research questions to examine the effect of the central unit and the effectiveness of autonomous team organization.
April 2023 - December 2023	Identify and analyze the case	Through reviewing the literature, I defined the RA$_2$ and RA$_3$. Accordingly, I examined two cases: in the first case, 8 employees were involved, while in the second case, 4 employees who met the criteria were included. The data collection method is summarized in Table 4. With the appropriate data, I parameterized the new SSPSP method, which was developed in Matlab environment.
January 2024 - May 2024	Evaluate solutions	During this period, I analyzed the simulation results using statistical methods and drew the relevant conclusions.
June 2024 - August 2024	Validate and verify the results	Using empirical data and participant observation methods, I re-measured the accuracy of the simulation. For the first case, 47 software developers were involved, while for the second case, 20 developers were involved in half-day teamwork, representing the structure of the simulation teams through the Marshmallow Challenge.

The participants had to meet the following conditions to take part in the study:

- they had participated in a DISC or Belbin training organized by an external company in the past 6 months, where their types were documented
- they had worked together on projects in the past 1 year, having at least 10 common tasks
- their hard skills could be measured using internal data from the past 1 year.

Table 4 summarizes the sources used during data collection.

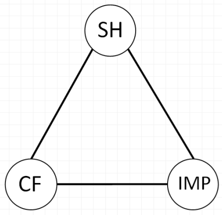
Table 4: Data Sources

Data Source	Details
Training material	8 training materials from the Belbin's training and 4 training materials from the DISC training were asked from the HR department to select the team members for the simulation where in both cases an external trainer made the results of the tests available. Additional 20 DISC training paper were asked for the validation process
Internal database	Internal database was used to measure hard skills in both cases
Questionnaire for the simulation	to measure the soft skills of the 8 different Belbin's team roles and 4 different DISC behavioral types a questionnaire was made by the HR department using 10 point Likert scale
Questionnaire of team roles	120 Belbin's self perception inventory questionnaire for determination of the Belbin's team roles of different project teams where 47 different Belbin's team roles were selected for the validation
Interviews	8 Belbin teams and 5 DISC teams were interviewed to report on their experience of validation teamwork

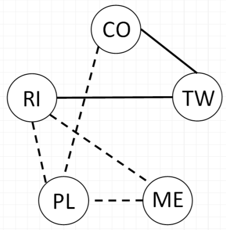
To answer RQ_2, the investigation was carried out within a project following the traditional project management approach, which is the common methodology in the Continental R&D department. Additional simulation data were provided through the data of 8 employees with different Belbin's team roles. In this case, I examined the impact of action-oriented groups on project success using the thoery of Belbin's team roles. Thus, I compared 8 different team structures through simulation, which are shown in Figure 5. In total, 115,200 cases were examined in the simulation. Non-parametric statistical analyses confirmed the significantly different performances of the various team structures. This allowed for comparing the cases when different Belbin's action-oriented group members joined the Belbin team's thinking-oriented and people-oriented groups. The simulation results answering RQ_2 are summarized in Figure 7 and Figure 8. A summary of the validation results can be found in Table 5.

The investigation of RQ_3 was carried out within the newly introduced agile projects methodology at the Continental R&D department, where I examined the effectiveness of autonomous team selection considering DISC behavioral types. The relevant project data for the simulation were determined based on one sprint of an agilely managed project. For the additional simulation data, I used the data of 4 individuals with different DISC behavioral types. Based on this, I identified one self-organized and 4 teams with dedicated leader (Figure 6). Thus, a total of 432,000 cases were examined in the simulation. Parametric statistical analyses confirmed the significantly different performances of the various team structures, which are summarized in Figure 10. A summary of the validation results can be found in Table 6.

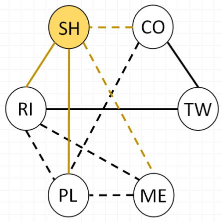
	PL	CO	TW	ME	IMP	RI	CF	SH
PL								
CO								
TW								
ME								
IMP							1,25	1,50
RI								
CF								1,25
SH								



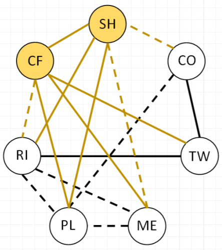
	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75		0,50		
CO			1,25	1,00		1,00		
TW				1,00		1,50		
ME						0,50		
IMP								
RI								
CF								
SH								



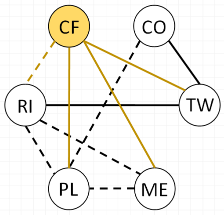
	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75		0,50		1,50
CO			1,25	1,00		1,00		0,50
TW				1,00		1,50		1,00
ME						0,50		0,50
IMP								
RI								1,50
CF								
SH								



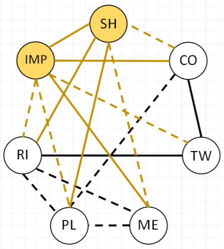
	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75		0,50	1,50	1,50
CO			1,25	1,00		1,00	1,00	0,50
TW				1,00		1,50	1,25	1,00
ME						0,50	1,50	0,50
IMP								
RI							0,50	1,50
CF								1,25
SH								



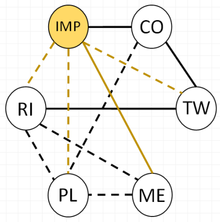
	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75		0,50	1,50	
CO			1,25	1,00		1,00	1,00	
TW				1,00		1,50	1,25	
ME						0,50	1,50	
IMP								
RI							0,50	
CF								
SH								



	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75	0,50	0,50		1,50
CO			1,25	1,00	1,50	1,00		0,50
TW				1,00	0,75	1,50		1,00
ME					1,50	0,50		0,50
IMP						0,50		1,50
RI								1,50
CF								
SH								



	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75	0,50	0,50		
CO			1,25	1,00	1,50	1,00		
TW				1,00	0,75	1,50		
ME					1,50	0,50		
IMP						0,50		
RI								
CF								
SH								



	PL	CO	TW	ME	IMP	RI	CF	SH
PL		0,50	1,00	0,75	0,50	0,50	1,50	
CO			1,25	1,00	1,50	1,00	1,00	
TW				1,00	0,75	1,50	1,25	
ME					1,50	0,50	1,50	
IMP						0,50	1,25	
RI							0,50	
CF								
SH								

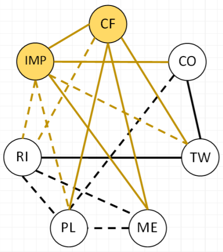


Figure 5:
Possible team structures of the examined Belbin's team roles
In the figure, negative synergistic relationships are represented by dashed lines, while positive synergistic relationships are shown with solid lines. The action-oriented group members, who join the existing Belbin team, as well as their newly established synergistic relationships, are marked with orange.

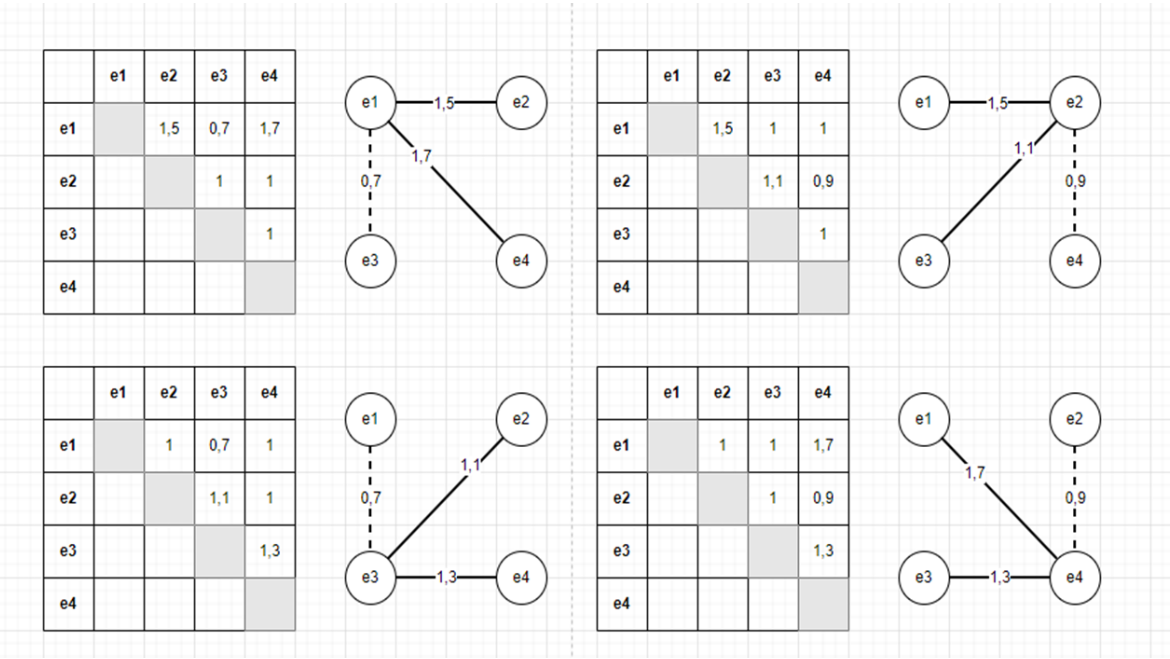


Figure 6:
Possible team structures of the examined DISC behavior types
Considering each DISC behavior types in this figure: : e1 - dominance, e2 - influence, e3 - steadiness, e4 - conscientiousness
AIn the figure, negative synergistic relationships are represented by dashed lines, while positive synergistic relationships are shown with solid lines.

Based on Figure 7, among the Belbin team structures, the lowest project time (TPT_{syn}) (53.6) and the lowest total project cost (TPC_{syn}) (80) were achieved by the action-oriented group (A team) when considering the synergetic effects. However, without considering the synergetic effects (Figure 8), the "B team + CF + IMP" group achieved the lowest project time (TPT_{nosyn}) (55.2), although the action-oriented group still had a better result for the total project cost (TPC_{syn}) (92.2). This essentially means that, without considering the synergetic effects, Belbin's team roles can be more successful by excluding the SH role if we only look at total project time. However, considering the synergetic effects, the action-oriented group (A team) is sufficiently efficient to complete a project. The worst values in all cases were achieved by the groups that included the SH role. However, where even one action-oriented group member joined the team, total project time decreased compared to when the action-oriented group member was not part of the team.

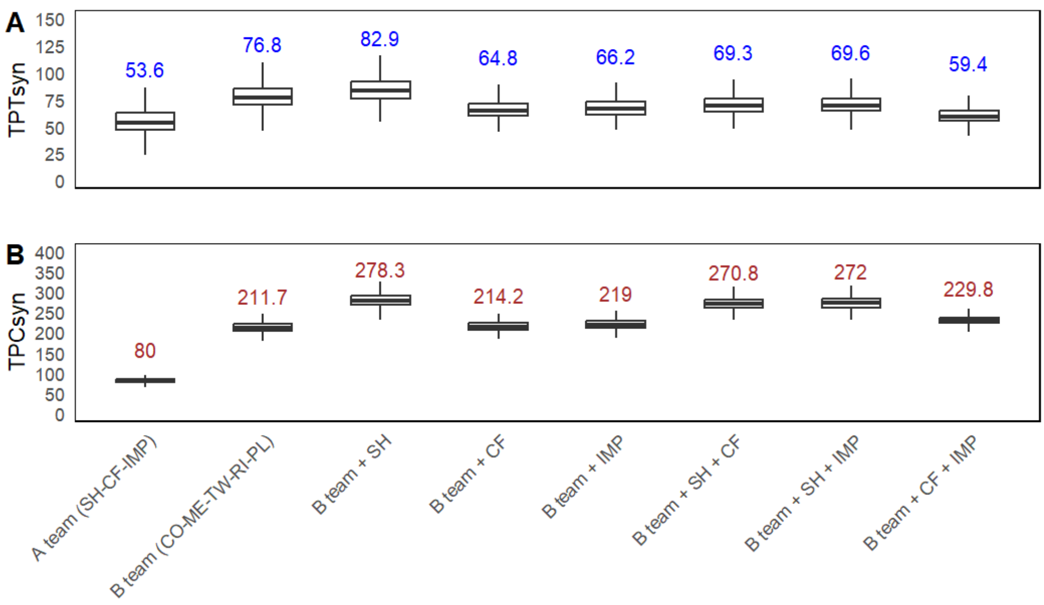


Figure 7: Order of the different team selections considering the synergies. **A** represents the TPT_{syn} and **B** represents the TPC_{syn} which each team achieves during the completion of the project.

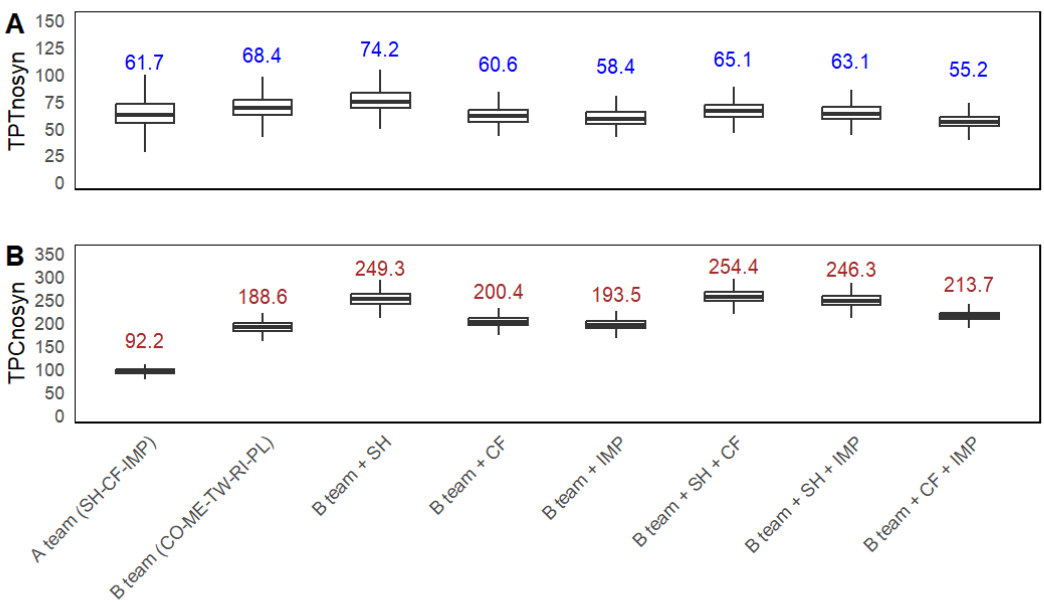


Figure 8: Order of the different team selections neglecting the synergies. **A** represents the TPT_{nosyn} and **B** represents the TPC_{nosyn} which each team achieves during the completion of the project.

Individual teams can already be compared based on Figure 7 and Figure 8, but it is important to highlight that the size of the teams differs (the number of members can vary from 3 to 8). Therefore, it is necessary to correct the previous comparison by dividing the results of the teams by the number of their members (normalizing). The TPT per team member is difficult to interpret, so in this case I only examined the TPC per team member (TPCsyn/n or TPCnosyn/n).

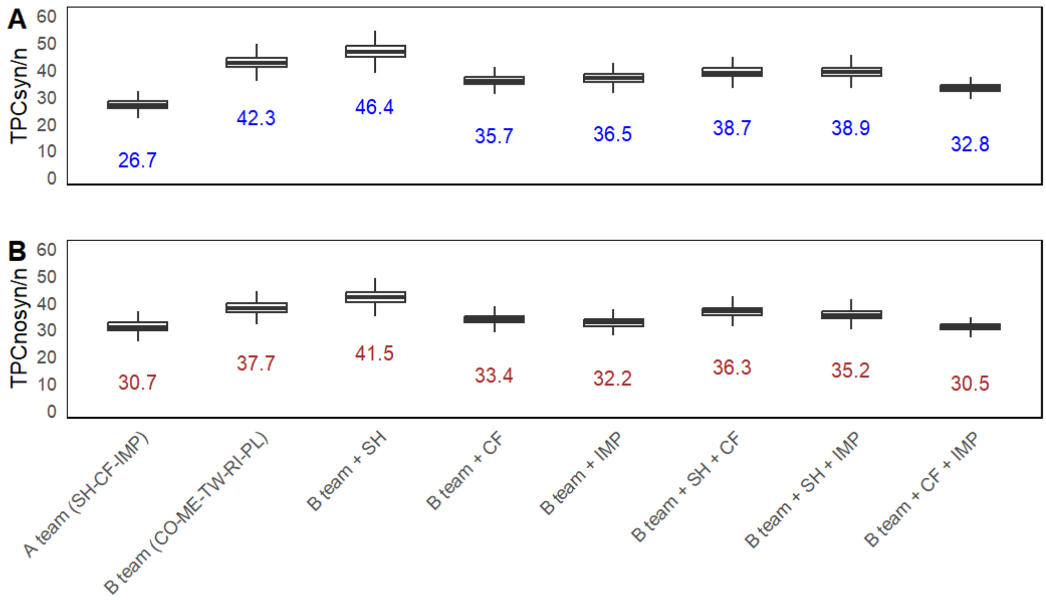


Figure 9: Order of the different team selections regarding the TPC of 1 person and considering (A) (neglecting (B)) the synergies.
Note: n is the number of persons in a group

Based on Figure 9, it can be concluded that the normalized project cost considering synergy (TPC_{syn}/n) was lowest for the action-oriented team (A team), while without considering synergy (TPC_{nosyn}/n), the "B team + CF + IMP" group achieved the lowest cost. In this regard, it can be said that, when considering synergetic effects, the action-oriented team is successful on its own. However, without synergetic effects, excluding the SH role is enough. This, of course, does not mean that the SH role can be excluded from software development project teams. The results assume an ideal world where there is no time pressure or unexpected effects. The biggest advantage of the SH role is that it can help the team navigate through unexpected and difficult situations while maintaining task focus.

Regarding autonomous team selection, team structures could only be compared based on the total project cost, as there were no significant differences in total project time. Figure 10 shows that the autonomous team achieved the best total project cost (TPC_{syn}) during the simulation. Additionally, a ranking can be established among the different DISC behavioral types as the leader of the teams. In small agile teams, it is advisable to choose a dominant (D) or conscientious (C) leader when autonomy (O) is not possible. Influence (I) or steadiness (S) behavior type leaders should be avoided. This is likely because, in an autonomous team, no central role emerges, so every team member can perform their task and complement each other based on their own abilities.

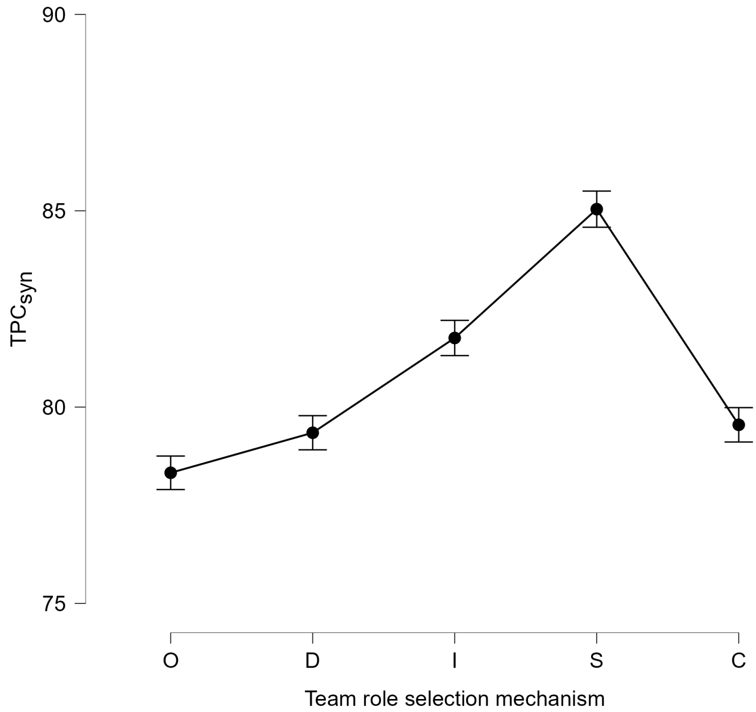


Figure 10: Project costs for various kinds of team role selection mechanisms where there is no constraint ($C_c\% = C_t\% = 1, C_s\% = 0$)

To confirm the simulation results, I designed a validation process using participant observation method. The essence of this process was to observe the behavior of the team structures examined during the simulation in a team game, where the human factors used in the simulation also played a crucial role. For this, I used the marshmallow challenge game, where I chose and grouped 47 individuals with different Belbin's team roles from a pool of 120 individuals who had filled out the Belbin's self perception questionnaire, as well as 20 selected individuals with different DISC behavior types. In both cases, the observation took half a working day, following the same pattern: (1st step) team-building, (2nd step) group discussion around a selected topic, dividing the total population into 4, (3rd step) half-hour guided introduction to teammates, and (4th step) the marshmallow challenge. Finally, the event concluded with a group reflection and feedback session. During the marshmallow challenge, I measured the duration of different teamwork activities and the height of the constructed towers. The results for examining the impact of the central action-oriented group in Belbin's team are summarized in Table 5, and the results for examining the impact of autonomous teams characterized by DISC behavior types are summarized in Table 6.

Table 5: Result of the validation of the central team role's effect on the team performance

Test type	A	B	C	D	E	F	G	H	Constraint
S: Average TPT	55	76	83	68	67	71	70	60	-
S: Rank	1	7	8	4	3	6	5	2	-
M: Time	8:36	14:25	15:32	12:34	12:53	13:50	13:45	11:42	max 14 min
M: Rank	1	7	8	3	4	6	5	2	-
M: Tower height	37 cm	28 cm	44 cm	30 cm	26 cm	38 cm	31 cm	31 cm	min 30 cm

Note. S: Simulation, *M*: Marshmallow game
Team structures according to the simulation: **A** = "A team (SH+CF+IMP), **B** = "B team (CO+ME+TW+RI+PL)", **C** = "B team + SH, **D** = "B team + IMP", **E** = "B team + CF", **F** = "B team + SH + IMP", **G** = "B team + SH + CF", **H** = "B team + CF + IMP"

Table 6: Result of the validation of autonomous team selection's effect on team performance

Test type	O	D	I	S	C	Constraint
Simulation: Rank	1	2	4	5	3	-
Marshmallow game: Time	6:32	8:50	12:54	13:45	13:30	max 14 min
Marshmallow game: Rank	1	2	4	5	3	-
Marshmallow game: Tower height	30 cm	40 cm	32 cm	32 cm	36 cm	min 30 cm

Abbreviations for team structures are the same as those used in the simulation, see Figure 10

The results of the marshmallow challenge confirmed the simulation results. Therefore, I conclude that the action-oriented group, as the central group in the synergetic network of Belbin's team roles, impacts the success of the project team. Furthermore, I conclude that during autonomous team selection, which was characterized by DISC behavior types, the autonomously formed team performs better than any other team with a dedicated leader of different behavior types. In addition, the validation results support the simulation outcomes, thus confirming the effectiveness of the newly developed project scheduling method. Based on both the simulation results and the validation results, I accept my research hypotheses RA_1, RA_2, and RA_3.

5 Research theses

Three research theses were formulated in alignment with the research questions, carefully considering both the simulation results.

RT_1 The proposed extended SSPSP method can incorporate Belbin team roles or DISC behavioral types by considering the synergies among these roles, as well as the soft and hard skills they represent, within a flexible or strict software environment. After appropriate hyper-parametrization, the method can also give reliable results on test projects.

RT$_2$ With the usage of the extended SSPSP the importance of the central team roles of the Belbin's team is proved, within the constraints and objective functions defined in the extended SSPSP. Although the presence of IMP and CF roles is certain, the presence of SH is controversial.

RT$_3$ With the usage of the extended SSPSP the positive impact of autonomous teams is proved, within the constraints and objective functions defined in the extended SSPSP.

6 Summary and Conclusion

6.1 Contribution to the literature

The research contributes to the project management literature from several perspectives. I presented a significantly more accurate software project scheduling method from a methodological point of view. In the first step, I identified the trend of the evolution of project management in the context of software development projects, which helped identify the most important current issues. In light of this, I systematically reviewed the existing literature on software project scheduling problems and identified gaps in the literature. Additionally, I identified two practical problems: the consideration of autonomous team selection in project scheduling and the effect of the central unit on the success of the project team. Addressing both of these practical issues is important for the success of software development projects. Based on the literature, I explored the possibility of modeling the two practical problems. Accordingly, during autonomous team formation, team structures that can be formed in different ways are characterized by DISC behavioral types, while the examination of the central group's effect is characterized by Belbin's team roles. In terms of the literature, I created synergy networks between the DISC behavioral types and Belbin's team roles.

1. With the results of the research, I contributed to the software project scheduling literature by developing a new method. I hyper-parameterized the method and then compared its accuracy with existing methods using a validated database. The results confirmed the accuracy of the new method, even with the limitations of the validated database.

2. Using the newly developed SSPSP method, I examined the impact of autonomous team selection and the central team unit on project success. For this, I created a case study based on empirical data, in which I first parameterized the newly developed SSPSP method based on real data and then validated the simulation results obtained in a real-life environment using the participant observation method. The results not only confirmed my research assumptions but also reinforced the effectiveness of the new SSPSP method.

6.2 Implications and Limitation

The extended new SSPSP method integrates several factors into the study of project team dynamics, thus providing a foundation for a new direction in software project scheduling. The method can be further developed by incorporating learning and forgetting curves, dynamic team formation theories, or taking into account multi-project and portfolio management considerations. The method has been validated, confirming its reliability and practical applicability, which could be particularly interesting for organizations facing people-oriented scheduling or human resource problems. One of the limitations of the method is that it is recommended only for short-term scheduling, as it does not consider longer-term interactions such as learning and forgetting, or team structure transformations.

Using the new method, I proposed scheduling for a team where the central unit of the team - thus the team structure or the team's capability - changes. This type of study has opened a new direction in the literature, namely the integration of dynamic capabilities into project scheduling. Since the scheduling proved accurate in the short term, the new method is recommended for any company where a central unit could emerge within already formed teams. For this, reliable capability and team role assessments are needed as input. Thus, the method could be used to determine the impact of everyday problems on scheduling, such as if a central employee is absent from team collaboration. The limitation of the research is that the effect of every team role on the success of project scheduling has not been examined, so further studies are needed in this direction. Furthermore, an important research direction could be the appearance of non-dominant team roles, which would necessarily rearrange the team structure.

Additionally, the method can be used to understand the functioning of autonomous teams as proposed by the agile mindset and to schedule tasks assigned to autonomous teams in a software project environment. A key limitation here is that during autonomous selection, the long-term effects were not considered.

In summary, the research achieved its goals, but it has several important limitations for practical application. The most important of these is that the new method is primarily suitable for short-term project planning and does not examine long-term factors. Additionally, several factors already modeled in the SSPSP literature were not considered to maintain the model's clarity and avoid complexity, so the results should be interpreted in the proper context.

6.3 Research Summary Table

I have summarized my research questions, research assumptions, and research theses formulated in light of the research results in Table 7.

Table 7: Research summary

Item	Statement
RQ1:	How can a software project scheduling method be enhanced to incorporate the uniqueness of different team roles and behavioral types, while also considering their interactions within a heterogeneous network, shaped by diverse skill sets and synergy effects, in both structured and flexible environments?
RA1:	The SSPSP method can be expanded to incorporate Belbin team roles and DISC behavioral types by leveraging the synergies among these roles, as well as the soft and hard skills they represent, within a flexible software environment.
RT1:	The proposed extended SSPSP method can incorporate Belbin team roles or DISC behavioral types by considering the synergies among these roles, as well as the soft and hard skills they represent, within a flexible or strict software environment. After appropriate hyperparameterization, the method can also give reliable results on test projects.
RQ2:	How do central team roles as the central unit of a heterogeneous network influence the success of software projects through their integration into scheduling strategies?
RA2:	The presence of central team roles in software projects positively impacts project success, thereby enhancing performance within the constraints and objective functions defined in the supplemented SSPSP.
RT2:	With the usage of the extended SSPSP the importance of the central team roles of the Belbin's team is proved, within the constraints and objective functions defined in the extended SSPSP. Although the presence of IMP and CF roles is certain, the presence of SH is controversial.
RQ3:	How does autonomously selected team as a heterogeneous network affect the success of software projects through their scheduling?
RA3:	Autonomous teams positively impact the success of software projects, within the constraints and objective functions of the enhanced SSPSP, more effectively than teams with dedicated leaders.
RT3:	With the usage of the extended SSPSP the positive impact of autonomous teams is proved, within the constraints and objective functions defined in the extended SSPSP.

7 The author's publications related to the topics

Journal Articles

1. Kosztyán, Z. T., **Harta, P.**, & Szalkai, I. (2024). The effect of autonomous team role selection in flexible projects. *Computers & Industrial Engineering*, 190, 110079. DOI:https://doi.org/10.1016/j.cie.2024.110079

2. Kosztyán, Z. T., **Harta, P.**, Szalkai, I., & Obermayer, N. (2024). Csoportkiválasztási módszerek vizsgálata rugalmas projekttervezési megközelítések esetén. *SZIGMA Matematikai-közgazdasági folyóirat*, 55(1), 53-83.
DOI: https://doi.org/10.15170/SZIGMA.55.1195

Proceedings

1. **Harta, P.**, Sebrek, S. S., Kosztyán, Z. T. (2025). Unveiling the Human Factor: How Personality Types Shape Software Project Scheduling and Synergy through Dynamic Managerial Capabilities – A Longitudinal Action Design Research. *19th Organization Studies Summer Workshop*, Chania, Crete, Greece, May, 22–24, 2025.

2. Kosztyán, Z. T., & **Harta, P.** (2023). Autonomous team role selection on flexible projects. In DS 126: *Proceedings of the 25th International DSM Conference (DSM 2023)*, Gothenburg, Sweden, October, 03-05, 2023 (pp. 019-028). DOI: https://doi.org/10.35199/dsm2023.03

Conferences

1. Harta, P. (2024). Központi szerepek hatása a szoftver projektek sikerességére. Abstract. XVIII. Gazdaságmodellezési Szakértői Konferencia, Pannon Egyetem, Veszprém, Hungary

2. Harta, P. (2024). Központi szerepek hatása a szoftver projektek sikerességére. Abstract. XXVII. Tavaszi Szél Konferencia, Óbudai Egyetem, Budapest, Hungary.

3. Harta, P., & Kosztyán Z. T. (2023). The effect of autonomous team role selection in flexible projects. The 25th International DSM Conference (DSM 2023), Chalmers University of Technology, Gothenburg, Sweden.

4. Harta, P. (2023). Examining the motivators of software developers during agile transformation. PMUNI workshop, Vienna, Austria.

5. Harta, P. (2023). Központi szerepek hatása a szoftver projektek sikerességére. Abstract. OGIK 2023, Pécsi Tudományegyetem, Pécs, Hungary.

6. Harta, P. (2023). Önszerveződő csapatok vizsgálata szoftver projekt környezetben. Abstract. XXVI. Tavaszi Szél Konferencia, Miskolci Egyetem, Miskolc, Hungary.
7. Harta, P., & Kosztyán Z. T. (2022). A személyiségtípusok szerepe a szoftver projektek tervezésében. Abstract. OGIK 2022, Salgótarján, Somoskő, Hungary.
8. Harta, P., & Kosztyán Z. T. (2022). The role of personality types in software project planning. PMUNI workshop, Corvinus University, Budapest, Hungary.
9. Harta, P. (2022). Kvantitatív módszerek kidolgozása a projekttervek készítésének támogatására. Abstract. XXV. Tavaszi Szél Konferencia, Pécsi Tudományegyetem, Pécs, Hungary.
10. Harta, P. (2021). Investigation of the optimal team selection in the agile software project environment. PMUNI workshop, Online Conference.
11. Harta, P. (2021). Agilis projektcsapat optimális kiválasztásának vizsgálata szoftver projekt környezetben. Abstract. OGIK 2021, Pannon Egyetem, Veszprém, Hungary.

References

Alabajee, M. A.-A., Ahmed, D. R., and Alreffaee, T. R. (2021). Solving software project scheduling problem using grey wolf optimization. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 19(6):1820–1829.

Alba, E. and Chicano, J. F. (2007). Software project management with {GAs}. *Information Sciences*, 177(11):2380–2401.

Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A., and Storey, M.-A. (2021). The diversity crisis in software development. *IEEE Software*, 38(2):19–25.

Aryanee, D., Razali, R., and Mansor, Z. (2020). Team formation for agile software development: a review. *Int. J. Adv. Sci. Eng. Inf. Technol*, 10:2088–5334.

Balcar, J. (2016). Is it better to invest in hard or soft skills? *The Economic and Labour Relations Review*, 27(4):453–470.

Bear, J. B. and Woolley, A. W. (2011). The role of gender in team collaboration and performance. *Interdisciplinary science reviews*, 36(2):146–153.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., et al. (2001). The agile manifesto.

Belbin, R. M. (1981). *Management Teams: Why They Succeed Or Fail*. Routledge.

Bell, S. T., Brown, S. G., and Weiss, J. A. (2018). A conceptual framework for leveraging team composition decisions to build human capital. *Human Resource Management Review*, 28(4):450–463.

Bell, S. T. and Outland, N. (2017). Team composition over time. In *Team dynamics over time*, pages 3–27. Emerald publishing limited.

Blazewicz, J., Lenstra, J. K., and Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1):11–24.

Borges, G. G. and de Souza, R. C. G. (2024). Skills development for software engineers: Systematic literature review. *Information and Software Technology*, 168:107395.

Cheng, J., Ji, J., Guo, Y.-n., and Ji, J. (2019). Dynamic multiobjective software project scheduling optimization method based on firework algorithm. *Mathematical Problems in Engineering*, 2019(1):8405961.

Cleland, D. I. (1994). *Project Management – Strategic Design and Implementation (2nd edn.)*. New York, McGraw-Hill.

Coelho, J. and Vanhoucke, M. (2011). Multi-mode resource-constrained project scheduling using rcpsp and sat solvers. *European Journal of Operational Research*, 213(1):73–82.

De Wit, A. (1988). Measurement of project success. *International journal of project management*, 6(3):164–170.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Diekmann, J. and König, C. J. (2018). Personality testing in personnel selection: Love it? leave it? understand it! In *Current issues in work and organizational psychology*, pages 17–31. Routledge.

Dingsøyr, T. and Dybå, T. (2012). Team effectiveness in software development: Human and co-operative aspects in team effectiveness models and priorities for future studies. In *2012 5th international workshop on co-operative and human aspects of software engineering (chase)*, pages 27–29. IEEE.

Dwertmann, D. J., Nishii, L. H., and Van Knippenberg, D. (2016). Disentangling the fairness & discrimination and synergy perspectives on diversity climate: Moving the field forward. *Journal of Management*, 42(5):1136–1168.

Dybå, T., Dingsøyr, T., and Moe, N. B. (2014). Agile project management. *Software project management in a changing world*, pages 277–300.

Galinsky, A. D., Todd, A. R., Homan, A. C., Phillips, K. W., Apfelbaum, E. P., Sasaki, S. J., Richeson, J. A., Olayon, J. B., and Maddux, W. W. (2015). Maximizing the gains and minimizing the pains of diversity: A policy perspective. *Perspectives on Psychological Science*, 10(6):742–748.

García-Nájera, A. and del Carmen Gómez-Fuentes, M. (2014). A multi-objective genetic algorithm for the software project scheduling problem. In Gelbukh, A., Espinoza, F. C., and Galicia-Haro, S. N., editors, *Nature-Inspired Computation and Machine Learning*, pages 13–24, Cham. Springer International Publishing.

Gilal, A. R., Jaafar, J., Basri, S., Omar, M., and Abro, A. (2016). Impact of software team composition methodology on the personality preferences of malaysian students. In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pages 454–458. IEEE.

Görög, M. (2003). *A projektvezetés mestersége*. Aula, Budapest.

Group, T. S. (2015). Chaos report. Technical report, Massachusetts.

Group, T. S. (2021). Chaos report: Beyond infinity. Technical report, Massachusetts.

Groysberg, B., Polzer, J. T., and Elfenbein, H. A. (2011). Too many cooks spoil the broth: How high-status individuals decrease group effectiveness. *Organization Science*, 22(3):722–737.

Guimera, R., Uzzi, B., Spiro, J., and Amaral, L. A. N. (2005). Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308(5722):697–702.

Gupta, A., Poels, G., and Bera, P. (2022). Using conceptual models in agile software development: a possible solution to requirements engineering challenges in agile projects. *IEEE Access*, 10:119745–119766.

Gutierrez, G., Garzas, J., de Lena, M. T. G., and Moguerza, J. M. (2018). Self-managing: an empirical study of the practice in agile teams. *IEEE software*, 36(1):23–27.

Hamilton, B. H., Nickerson, J. A., and Owan, H. (2012). Diversity and productivity in production teams. In *Advances in the Economic Analysis of participatory and Labor-managed Firms*, pages 99–138. Emerald Group Publishing Limited.

Hartmann, S. and Briskorn, D. (2022). An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of operational research*, 297(1):1–14.

Hegazy, T., Shabeeb, A. K., Elbeltagi, E., and Cheema, T. (2000). Algorithm for scheduling with multiskilled constrained resources. *Journal of construction engineering and management*, 126(6):414–421.

Hidayati, A., Budiardjo, E. K., and Purwandari, B. (2020). Hard and soft skills for scrum global software development teams. In *Proceedings of the 3rd International Conference on Software Engineering and Information Management*, pages 110–114.

Hoda, R., Noble, J., and Marshall, S. (2010). Organizing self-organizing teams. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pages 285–294.

Hoda, R., Noble, J., and Marshall, S. (2012). Self-organizing roles on agile software development teams. *IEEE Transactions on Software Engineering*, 39(3):422–444.

Jehn, K. A., Northcraft, G. B., and Neale, M. A. (1999). Why differences make a difference: A field study of diversity, conflict and performance in workgroups. *Administrative science quarterly*, 44(4):741–763.

Kaliprasad, M. (2005). Agile project management: How to succeed in the face of changing project requirements. *Cost Engineering*, 47(10):29.

Kanski, L., Budzynska, K., and Chadam, J. (2023). The impact of identified agility components on project success—ict industry perspective. *Plos one*, 18(3):e0281936.

Kosztyán, Z. T., Bogdány, E., Szalkai, I., and Kurbucz, M. T. (2022). Impacts of synergies on software project scheduling. *Annals of Operations Research*, 312:883–908.

Lee, D., Huh, Y., and Reigeluth, C. M. (2015). Collaboration, intragroup conflict, and social skills in project-based learning. *Instructional science*, 43:561–590.

Li, H., Zhu, H., Zheng, L., and Liu, Y. (2023). Software project scheduling with multitasking. *Economic Computation & Economic Cybernetics Studies & Research*, 57(1).

Liemhetcharat, S. and Veloso, M. (2012). Modeling and learning synergy for team formation with heterogeneous agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 365–374.

Lundin, R. A. and Söderholm, A. (1995). A theory of the temporary organization. *Scandinavian Journal of management*, 11(4):437–455.

Lykourentzou, I., Antoniou, A., Naudet, Y., and Dow, S. P. (2016). Personality matters: Balancing for personality types leads to better outcomes for crowd teams. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 260–273.

Maghsoudlou, H., Afshar-Nadjafi, B., and Niaki, S. T. A. (2016). A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. *Computers & chemical engineering*, 88:157–169.

Marston, W. M. (1928). Review of a short outline of comparative psychology. *Journal of Abnormal and Social Psychology*, 23(2):256–257.

Matturro, G. (2013). Soft skills in software engineering: A study of its demand by software companies in uruguay. In *2013 6th international workshop on cooperative and human aspects of software engineering (CHASE)*, pages 133–136. IEEE.

Meckenstock, J.-N. (2024). Shedding light on the dark side–a systematic literature review of the issues in agile software development methodology use. *Journal of Systems and Software*, page 111966.

Meier, A. and Kock, A. (2023). The human factor in agility: Exploring employee dedication in agile project organizations. *International Journal of Project Management*, 41(7):102527.

Meslec, N. and Curşeu, P. L. (2015). Are balanced groups better? belbin roles in collaborative learning groups. *Learning and Individual Differences*, 39:81–88.

Myszkowski, P. B., Laszczyk, M., and Lichodij, J. (2017). Efficient selection operators in nsga-ii for solving bi-objective multi-skill resource-constrained project scheduling problem. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 83–86. IEEE.

Myszkowski, P. B., Laszczyk, M., Nikulin, I., and Skowroński, M. (2019). imopse: a library for bicriteria optimization in multi-skill resource-constrained project scheduling problem. *Soft Computing*, 23:3397–3410.

Myszkowski, P. B., Skowroński, M. E., Olech, Ł. P., and Oślizło, K. (2015). Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. *Soft Computing*, 19:3599–3619.

Myszkowski, P. B., Skowroński, M. E., and Podlodowski, Ł. (2013). Novel heuristic solutions for multi-skill resource-constrained project scheduling problem. In *2013 federated conference on computer science and information systems*, pages 159–166. IEEE.

Peslak, A. R. (2006). The impact of personality on information technology team projects. In *Proceedings of the 2006 ACM SIGMIS CPR conference on computer personnel research: Forty four years of computer personnel research: achievements, challenges & the future*, pages 273–279.

Phillips, K. W., Liljenquist, K. A., and Neale, M. A. (2009). Is the pain worth the gain? the advantages and liabilities of agreeing with socially distinct newcomers. *Personality and Social Psychology Bulletin*, 35(3):336–350.

Pieterse, V., Leeu, M., and van Eekelen, M. (2018). How personality diversity influences team performance in student software engineering teams. In *2018 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–6. IEEE.

Reiff, J. and Schlegel, D. (2022). Hybrid project management–a systematic literature review. *International journal of information systems and project management*, 10(2):45–63.

Reynierse, J. H., Ackerman, D., Fink, A. A., and Harker, J. B. (2000). The effects of personality and management role on perceived values in business settings. *International Journal of Value-Based Management*, 13:1–13.

Rezende, A. V., Silva, L., Britto, A., and Amaral, R. (2019). Software project scheduling problem in the context of search-based software engineering: A systematic review. *Journal of Systems and Software*, 155:43–56.

Salo, O. and Abrahamsson, P. (2008). Agile methods in european embedded software development organisations: a survey on the actual use and usefulness of extreme programming and scrum. *IET software*, 2(1):58–64.

Schulze, A. and Brusoni, S. (2022). How dynamic capabilities change ordinary capabilities: Reconnecting attention control and problem-solving. *Strategic Management Journal*, 43(12):2447–2477.

Schwaber, K. and Beedle, M. (2001). *Agile software development with Scrum*. Prentice Hall PTR.

Scullard, M. and Baum, D. (2015). *Everything DiSC Manual*. John Wiley & Sons.

Shenhar, A. J. and Dvir, D. (2007). Project management research—the challenge and opportunity. *Project management journal*, 38(2):93–99.

Söderlund, J. (2004). Building theories of project management: past research, questions for the future. *International journal of project management*, 22(3):183–191.

Towry, K. L. (2003). Control in a teamwork environment—the impact of social ties on the effectiveness of mutual monitoring contracts. *The Accounting Review*, 78(4):1069–1095.

Tuckman, B. W. (1965). Developmental sequence in small groups. *Psychological bulletin*, 63(6):384.

Van Knippenberg, D., De Dreu, C. K., and Homan, A. C. (2004). Work group diversity and group performance: an integrative model and research agenda. *Journal of applied psychology*, 89(6):1008.

Vega-Velázquez, M. Á., García-Nájera, A., and Cervantes, H. (2018). A survey on the software project scheduling problem. *International Journal of Production Economics*, 202:145–161.

VersionOne (2024). 17th-annual-state-of-agile-report. Technical report, VersionOne Inc.

Vishnubhotla, S. D., Mendes, E., and Lundberg, L. (2018). Designing a capability-centric web tool to support agile team composition and task allocation: a work in progress. In *Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering*, pages 41–44.

Waleed, A., Ahmad, W., Jan, M., Ali, S., Khattak, A., and Nadeem, A. (2021). The effect of team value diversity on team performance: The mediating role of relationship conflict and the moderating effects of organization citizenship behavior and leader-member exchange quality. *Management Science Letters*, 11(8):2185–2194.

Wankhede, V. A. and Vinodh, S. (2021). Analysis of industry 4.0 challenges using best worst method: A case study. *Computers & Industrial Engineering*, 159:107487.

Wawak, S. and Woźniak, K. (2020). Evolution of project management studies in the xxi century. *International Journal of Managing Projects in Business*, 13(4):867–888.

Wysocki, R. K. (2010). *Effective software project management*. Wiley+ ORM.

Wysocki, R. K. (2019). *Effective project management: traditional, agile, extreme*. John Wiley & Sons.

Xiao, J., Ao, X.-T., and Tang, Y. (2013). Solving software project scheduling problems with ant colony optimization. *Computers & Operations Research*, 40(1):33–46.

Yin, R. K. (2009). *Case study research: Design and methods*, volume 5. sage.

Zainal, P., Razali, D., and Mansor, Z. (2020). Team formation for agile software development: a review. *Int. J. Adv. Sci. Eng. Inf. Technol*, 10(2):555–561.

Zapotecas-Martínez, S., García-Nájera, A., and Cervantes, H. (2020). Multi-objective optimization in the agile software project scheduling using decomposition. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pages 1495–1502.