

Facilitating the Decentralisation of Software Development Projects from a Project Management Perspective: A Literature Review

Sven Timmermann, Daniel Staegemann^a, Matthias Volk^b, Matthias Pohl^c, Christian Haertel, Johannes Hintsch and Klaus Turowski

*MRCC VLBA, Otto-von-Guericke University Magdeburg, Universitaetsplatz 2, 39106 Magdeburg, Germany
{sven.timmermann, daniel.staegemann, matthias.volk, matthias.pohl, christian.haertel, johannes.hintsch,*

Keywords: Decentralisation, Software Development, Project Management, Literature Review, Guidelines.

Abstract: With the increasing relevance of decentralisation for the software development process, being aware of the possible challenges and corresponding solutions has become more relevant than ever. The scientific body of knowledge is currently containing many publications about specific aspects of decentralisation, but is lacking in collections that cover more than one area. In this work, the challenges of different forms of decentralisation are examined by means of a literature review. Subsequently, the findings are evaluated and summarised into guidelines that can be applied by project managers and development teams to increase the success of decentralisation of the software development process.

1 INTRODUCTION

From a project management perspective, two trends are characteristic of software development projects in recent years: the turn to agile methodologies (Jamous et al. 2021) and the increasing decentralization. While many firms in the field now decentralise their projects by working in teams that are distributed across long distances (Casey and Richardson 2006), other forms of decentralisation are becoming more frequent as well. Both, the organisational structure and the ways of communication are affected by this trend. Several reasons led to an increase in decentralisation. For example, lower costs of development resulted in an increase of offshoring endeavours (Aspray et al. 2006). The spread of the COVID-19 pandemic starting in 2020 further increased the prevalence of decentralisation (Contreras et al. 2020; Pakos et al. 2021). The subsequent trend of working from home being encouraged or even made mandatory increased distribution and its challenges in many projects, not only in the software development field. Decentralising different parts of the process strongly differs in impact. Decentralising communication or knowledge may only include a change in

management of that aspect (Hellström et al. 2001), while decentralising the location often involves different cultures, which can affect all parts of the project landscape as discussed in (Krishna et al. 2004). As any changes of methods with such a wide range of affected processes and areas, decentralisation results in both benefits and challenges. With the relevancy of decentralisation as a concept, knowing common challenges and ways to face or avoid them becomes a crucial factor. Without appropriate awareness of these challenges and solutions for them, the benefits of decentralisation can often be overshadowed by the negative consequences of unsolved problems (Lanaj et al. 2013).

The current scientific body of knowledge already provides increasingly large amounts of studies related to this topic (Jiménez et al. 2009; Da Silva et al. 2010), yet there are very little studies that collect the challenges of different areas and offer methods and approaches for all aspects of decentralisation in one concentrated place. A collection of such guidelines that can be applied to real-life contexts can be a substantial contribution, as it allows its use outside or without the execution of studies prior to the application to development projects.

^a <https://orcid.org/0000-0001-9957-1003>

^b <https://orcid.org/0000-0002-4835-919X>

^c <https://orcid.org/0000-0002-6241-7675>

To find these working guidelines, the following research question (RQ) will be answered by means of a literature review:

RQ: What guidelines can be used to facilitate decentralisation of software development projects and successfully overcome its challenges?

To provide an answer to the overarching RQ, the following two sub research questions (SRQ) are discussed:

SRQ1.1: What are the challenges that are commonly faced in decentralised software development projects?

SRQ1.2: What are approaches and solutions for the identified challenges?

After this introduction into the relevance and motivation behind the topic examined as well as the researched questions, the used methodology is explained. In the third section, the findings of the literature review are presented and the research questions are answered in form of guidelines. Finally, a conclusion is given and possible directions for future studies are presented.

2 METHODOLOGY OF THE LITERATURE REVIEW

To establish a knowledge base for the research while simultaneously gauging the current state of the art in the field, a review process largely based on the approach of Levy and Ellis (2006) was used. The authors describe a structured review process that was altered slightly to fit the scope and context of this work.

The search process consisted of three steps that would be iteratively repeated as necessary.

Step 1: An initial keyword search that served as the starting point for the other two steps.

Step 2: A backwards reference search of the articles uncovered in step one.

Step 3: A forward reference search of articles found in both previous steps.

The keywords were used in different combinations and comprised the initial words used in the first search, as well as those found in previously identified research articles. The resulting list of keywords and phrases used is as follows: *decentralisation, work, communication, organisation, software development, project management, self-management, knowledge, flat organisation, hierarchy, success factors, agile software development.*

Both backwards and forwards searches were conducted across multiple levels, meaning that references found in identified articles were also considered. The backwards search was aimed at uncovering high quality foundational literature, that other high quality articles were based on, while the forward search was used to find “[...]follow-up studies or newer developments related to the phenomenon under study.” (Levy and Ellis 2006, p. 191)

Given the purpose and context of the review as the basis for a future case study (see Turnbull et al. 2021 for guidelines), the scope of the review process naturally had to be limited in comparison to the pure literature review articles. Because of this, the coverage as categorized by Cooper (1988) in his taxonomy, was only representative. An exhaustive coverage of the topic would warrant its own separate endeavour. A large number of articles had to be excluded from consideration during the search, which results in the need to systematically evaluate the relevancy of articles (vom Brocke et al. 2009). For this evaluation, mainly the titles and abstracts of articles in question were analysed. In some cases, however, the text of the articles was also examined, when the abstracts were not sufficient to reasonably exclude or include them. To increase overall quality of the literature used, journal articles were prioritised.

3 FINDINGS OF THE LITERATURE REVIEW

The concepts used to structure the review have been established after an initial scan of a portion of the chosen literature, to allow for a clear meaningful synthesis. For the literature review the concepts were derived partly from frequently mentioned keywords such as for example “Hierarchy” or “Knowledge Distribution”. In the other cases, concept names were manually formulated to encompass all included topics. Among others, this includes “Staffing Decentralised Teams” and “Communication Tools”. The results will be discussed by concept and within them in challenge-solution pairs, rather than presenting every challenge of one concept before moving on to approaches. However, if multiple problems with the same solution or multiple approaches to the same problem are mentioned across the articles, they are still presented together.

The found and reviewed literature covers the topic of decentralisation in a very general manner. Many articles that were found using the stated methods

cover forms of decentralisation that are unrelated to software development or any kind of development environments for that matter. When only examining decentralisation that is related to organisations or projects, there are also a large number of publications addressing other fields. Because of the applicability to the present study, the latter were considered for the review. To provide a better structure and comprehensibility, the concept of decentralisation as a whole is in the following divided into three sub-concepts, namely *organisational structure*, *communication*, *location* and the *process of decentralisation*.

3.1 Organisational Structure

Decentralisation of an organisational structure means to move towards a network of (self-managed) teams rather than a single team controlled by one leader, which brings, however, several challenges

3.1.1 Team Belonging

Any form of workplace regardless of field is affected by social influences. The social structure the workplace provides is equally important to employees as functional socialisation of team members is to projects. Especially in team-focused work such as software development, the concept of belonging to one or more teams is increasingly relevant. Various benefits of proper socialisation and feeling of belonging to a certain group are presented by different articles. Beehr et al. (2000) describe how positive group membership can decrease work stress, as the feeling of available support by co-workers reduces anxiety regarding problems. Additionally, they state that in such situations, the support can increase productivity as well as facilitate achieving company goals. Adler et al. (2008) name the increased trust between employees as a reason for improvements in cooperation. Lastly, in (Adler 2001) it is stated that a sense of community can lead to better performance regarding knowledge creation, which faces an increasing demand according to them.

As the literature shows, belonging to multiple teams as well as being spatially separated from team members can negatively impact the socialisation and therefore remove the named benefits. In (Marshall et al. 2007), a study of workplace isolation is provided, describing different factors and causes for the problem, many of which can be caused or amplified by decentralisation. According to them, workplace isolation is caused mostly by a perceived lack of essential positive factors. While this does not necessarily mean they are actually missing, this

includes availability of support by both co-workers and supervisor, opportunities for meaningful social interactions, inclusion in groups and its activities as well as recognition for performance and achievements. Especially the high number of actors described in “Multi-Team Systems” as well as the operation from co-workers caused by decentralisation of location as discussed in “Virtual Teams” can severely decrease perception of these factors, and this provoke workplace isolation. Especially with the introduction of physical distance and other drawbacks of decentralisation, the feeling of isolation is named as a challenge by articles such as (Mann et al. 2000) and (Pinsonneault and Boisvert 2001). Lastly, the positive effects of team belonging do not always translate to all aspects of decentralisation. As (Tajfel 1982) describes, positive team belonging and the consequent identification with the team’s goals and values can introduce a high level of competition and rivalry between teams, which in turn negatively influences multi-team systems through decreased cooperation and increased conflicts between teams according to (Lanaj et al. 2013).

Guideline 1: Employees should be assigned to only one team at a time whenever possible. Team building efforts should be performed to increase the benefits of team belonging for the team members that cooperate with each other in reality rather than in theory.

3.1.2 Multi-Team Systems

This concept, exclusive to the literature about decentralisation, covers the challenges in cooperating across multiple separate teams that work on the same project. This can but does not have to include different component teams, shared leaders between teams, and distribution across sites. The trend of splitting larger projects into smaller teams as part of decentralising organisational structures is covered. The most prevalent of the challenges associated with cooperation between multiple teams is the dramatic increase in coordination requirements for such systems. Many articles including (Leavitt 2005), (Magee and Galinsky 2008) and (Lundberg and Thompson 1967) mention coordination failures caused by decentralisation, especially regarding large scale organisations. A number of possible reasons causing these failures are described by (Lanaj et al. 2013). According to them, the high number of actors involved in such large-scale systems can directly hinder communication between all the different team members. Additionally, they state that the coordination of teams themselves can be impacted because of two reasons.

Firstly, if the same leaders are coordinating multiple teams of the same system, they cannot do so simultaneously, but are forced to coordinate one team first. This of course can mean that important information from a later team can be missing from the previous team's coordination.

Secondly, increasing flexibility inside of one team makes the coordination more difficult as the other teams consequently have more uncertainty regarding that teams' actions.

Smaller teams and organisational decentralisation can also lead to higher risk in development projects because these smaller teams tend to set higher, more difficult goals, which involve higher risks (Lanaj et al. 2013). This effect is even further increased in multi-team systems, as the need for high team performance can lead to a sense of rivalry between teams of the same system, resulting in even more risk seeking (Kilduff et al. 2010). This is especially problematic, since in multi-team systems risk is amplified according to a productive function rather than an additive one (Tversky and Kahneman 1983), increasing negative effects of the risk itself.

Lastly, all challenges to multi-team systems should be considered carefully, since according to (Lanaj et al. 2013) the failures of one team can lead to the failure of the whole system.

Guideline 2: When managing project risk in multi-team systems, the impact of the approach should be considered as to not underestimate the real risk that is present.

3.1.3 Hierarchy

The decentralisation of hierarchy is mainly discussed in two forms, reducing the number of hierarchical levels and effectively bringing all employees closer to the same level or alternatively handing control downwards in the hierarchy (Lee and Edmondson 2017). The resulting challenges are presented in a more general manner, grouping both perspectives together. Decentralising hierarchy in any form is associated with improvements in employee satisfaction and engagement regarding their work (Cohen and Ledford 1994). Still, (Lee and Edmondson 2017) warn that these benefits or overly positive reports can be misleading, as there is also evidence such as presented in (Barker 1993), where it is argued that prolonged work in a decentralised hierarchy, specifically under self-management, can lead to increased stress and may cause burnout. Lanaj et al. (2013) describe an increase in risk seeking behaviour that is based on the possibly exaggerated positive expectations self-managed teams develop,

which results in increased project risk. According to (Lee and Edmondson 2017), management efforts do not disappear together with the role of managers during decentralisation as the needed managerial tasks are distributed between team members. This can be problematic given that not every employee is properly trained or prepared to perform them. In (Bernstein et al. 2016) it is further described that not all people are equally drawn to or compatible with the self-managed organisational design, further complicating the process of decentralising hierarchy. Lastly, other project characteristics may influence the compatibility with a decentralised hierarchical approach. According to (Vallon et al. 2013), self-management is very difficult if the team is distributed or virtual, leading to the assumption that flattening hierarchy is better suited for co-located teams.

3.2 Communication

The decentralisation of communication is geared towards communicating via a network instead of one central point that redistributes information. This poses on the hand the question of tooling and on the other hand necessitates an effective way of distributing knowledge.

3.2.1 Communication Tools

As is the case with any tool usage, the choice of communication tooling has to be adjusted to the needs of any given project. Since communication is critical in decentralised and especially distributed projects and organisations, the most common discussion regarding tools involves those that are meant to facilitate it. They are often listed alongside of other tools used to improve cooperation, which will be partly presented here if they are not more relevant to another concept.

Firstly, Daft and Lengel (1986) already argued for care in one's choice of communication channel. They describe that well understood information should be communicated over a less formal channel, while ambiguous information should be transferred using standardised and rich channels. This also applies to tools that provide these communication channels. For example, Mäki et al. (2004) discuss the different use cases of e-mail and voice calls. According to them, e-mails are better suited for reaching multiple people, addressing more complex topics or sharing documents, while directly calling a colleague is the better option in cases where less complex topics have to be discussed immediately. They also present drawbacks of e-mail communication such as e-mails

not reaching the correct people, or the overload high amounts of e-mails can cause. Therefore, other tools should be used to compliment this communication channel. Jiménez et al. (2009) describe the need for cooperative tooling in the field of knowledge transfer. They argue that a tool, which allows for simultaneous work on diagrams and models would benefit the communication of complex processes or system information. Such a tool is for example presented in (Sarkar et al. 2008). Technical staff can use it to render structures, systems and architecture of applications in multiple languages, making it especially relevant for offshore cooperation.

In general, tools for cooperation and communication gain increased relevance when working in a distributed or even offshore setting according to (Vallon et al. 2013). Casey and Richardson (2006) argue that especially video conferencing is a critical factor in successfully executing distributed development, as things such as language barriers and misinterpreting who is talking can be significantly more impactful without a visual component. Cased and Richardson (2006) also describe additional challenges regarding tooling when cooperating across borders. They state that it is crucial for any distributed cooperation to use the same compatible tools across different sites, which can sometimes be challenging as support and warranties may apply differently for the same products when working in different countries. Therefore, even when the chosen tool fits the needs of a project optimally, the provided support should be considered, as the failure of such tools can be extremely problematic.

In a distributed setting, Jiménez et al. (2009) argue that it is often necessary to recentralise knowledge to ensure its availability to all members that might need it for their work. This can be done in form of centralised documentation or knowledge databases, for which they also stress the need to be updated constantly to provide their benefits. Zhuge (2002) suggests the use of an information repository that is created and updated by using communication tool records. Because any form of centralised knowledge base that holds all relevant project information will always be relatively large in size and will contain many different forms of information, Mohan and Ramesh (2007) present a traceability framework that is supposed to allow users too easily locate and identify key knowledge for their own processes. This is especially relevant as (Mäki et al. 2004) names problems in locating and accessing knowledge in such databases as their major drawbacks. Finally, Cased and Richardson (2006) warn that providing powerful communication tools

does not guarantee effective and meaningful usage of them. They argue that training and motivation measures are important to make use of the various tools' benefits.

Guideline 3: For development in decentralised project settings, complement or even replace e-mail communication with better-suited communication tools such as tools that include chatroom functionalities or video conferencing while also paying attention to compatibility across the organisation as well as the necessary user training.

Guideline 4: In decentralised projects, knowledge should be centralised using central documentation or a knowledge database including functionalities to effectively navigate and search for key information within it.

3.2.2 Knowledge Distribution

When viewed from a decentralisation perspective, knowledge distribution can describe both the decentralisation of knowledge as well as consequences of decentralisation for the knowledge distribution. The literature regarding this topic often describes the more general term knowledge management, which knowledge distribution is a part of. Decentralising the knowledge by distributing it across the team rather than having a single individual hold the entire knowledge about processes and other project information, is beneficial because the immense amount of information in software development projects is too expansive for a single project member (Lee and Edmondson 2017). Regardless, decentralisation may force a certain centralisation of knowledge as stated in (Mäki et al. 2004), which is additionally problematic because the employees holding key information are often people in key positions, resulting in them being less available for knowledge transfer.

Generally, to avoid overloading single team members with information, especially developers that work on specific components or parts of the project, Jiménez et al. (2009) suggest the use of a system or processes that notify team members when and only when changes occur that are relevant to them or their work. Cramton (2001) describes that distributed teams often face difficulty in upholding a mutual understanding of and knowledge about their shared work. Similarly, the problem reduced awareness regarding activities of other team members typically regarding coordination also translates to project knowledge.

As stated by Mäki et al. (2004), decentralisation can make roles and responsibilities unclear, which consequently reduces awareness of who holds what kind of important knowledge. They argue that this, together with the added difficulties in availability of team members and communication, results in an overall decrease in knowledge accessibility throughout decentralised projects. Paasivaara et al. (2009) suggest that agile approaches such as the scrum framework can be used to combat these awareness and communication deficits, as the regular short stand-up meetings that are used in agile contexts allowed for an increased transparency as well as an overview over the team's activities.

Different aspects or implications of decentralisation are listed in (Jiménez et al. 2009) as reasons for decreased knowledge transfer effectiveness. According to them, large networks, complex infrastructure, misunderstandings caused by unstandardised communication across many channels and tools as well as high response times together result in reduced communication frequency and quality. According to (Babar et al. 2006), a wide variety of cooperation tools can be used to address the named issues and can avoid ambiguity. The choice of processes and communication activities in general need to be adjusted to each individual team and project to reflect their specific needs (Maznevski and Chudoba 2000). Additionally, it should be noted that communication often involves large time investments, creating the need to further adjust what communication is needed and effective.

Guideline 5: Knowledge should be sparsely distributed to increase focus and avoid overloading individual team members with information. To make use of its benefits as well as avoid unawareness of where knowledge is located, communication tools, regular meetings and a bridgehead role that holds meta knowledge can be employed. The bridgehead role should be dedicated and “full time” to guarantee availability and effective knowledge distribution.

3.3 Location

Decentralisation of location means to move from co-located teams to projects that are executed across different sites or even countries. However, there are several approaches to achieve this.

3.3.1 Offshore

Cooperating with offshore co-workers does not only introduce large distances to the development team,

but also requires the teams to overcome cultural differences. The literature discusses these issues often as a topic related to but distinct from virtual teams, as cooperating with offshore partners guarantees team distribution and comes with its own separate set of challenges. The first challenge described by Krishna et al. (2004) comes in form of the choice of the project to decentralise as well as what offshore partner to cooperate with. They argue that projects are more or less applicable to offshore development based on their communication needs.

A lower need for communication (for example because the kind of product is well understood by both cultures) can make the process significantly easier. According to them, certain cultures can be closer in terms of their values and approaches, while other combinations of cultures do not necessarily work well together. Therefore, the choice of the country to cooperate with can be crucial.

Krishna et al. (2004) also mention that the differences in infrastructure are especially impactful because of the reliance on telecommunication in distributed work. As discussed in their respective sections, both multi-team systems and virtual or distributed teams tend to result in a higher number of involved actors. Since cooperating with an offshore team in many cases includes both, the relevance of team size, as described by Casey and Richardson (2006), becomes a critical factor.

They argue that if the offshore team is too large in comparison to the local staff on the project, the local team members are likely to face demotivation and fears of being replaced by their less expensive co-workers, which can lead to experienced and valuable employees leaving the company. This is especially problematic, when time zone differences make the coordination and training of a huge number of offshore colleagues more difficult because it severely limits the available time to synchronize.

The training of the offshore team members is another challenge (Casey and Richardson 2006), since the low cost of labour often comes with an inexperienced workforce that is less knowledgeable about the commonly used processes. They also argue that even though this is the case and local team members are often more comfortable in the existing local processes, they should not be applied to offshore cooperation without adjusting them to the new context. Casey and Richardson (2006) also describe the importance of giving process ownership to the people closest to the process, which often are the offshore developers.

Additionally, to the large distances between teams, differences in culture are another critical factor

in any offshore cooperation. Carey (1998) describes that in the same way language differences can lead to translation errors, cultural differences in terminology and problem understanding and solving can facilitate misunderstandings between teams and their members, which is supported by (Holmstrom et al. 2006). Casey and Richardson (2006) argue that additionally to conflict resolution, the culture can impact things such as how work time is perceived. According to them, these problems can result in valuable employees leaving projects or even companies if they are not addressed. Carey (1998) suggests the use of codification and translation guidelines to avoid misunderstandings.

Another broader approach discussed is the implementation of cultural trainings, which would be performed before the start of the cooperation (Forster 2000). Krishna et al. (2004) warn that such training endeavours are often done only to prepare offshore partners for cooperation, which they describe as problematic. Instead, trainings should be performed for both sides of the project team.

Similar to the trend to culturally train only the offshore team members, Krishna et al. (2004) also describe that a common challenge is the misguided attempt to only adjust the cooperating offshore teams to a local work culture. They instead suggest the use of cultural bridgeheads, team members that might work on the partner's premises and have experience regarding the partner's culture, allowing them to translate between the different cultures. According to Brannen and Salk (2000), the most effective approach would be to establish a separate, compromising work culture in order to create a work environment that is equally accessible to and compatible with all team members.

Guideline 6: To facilitate offshore cooperation, measures such as team building efforts, cultural training and processes adjustments should be implemented bilaterally in order to bring the cooperating teams closer together instead of attempting to adjust offshore teams to local standards.

Guideline 7: In cooperation involving different cultures, the bridgehead role gains importance and should be filled with a team member that has experience in working in and with the partner's culture, allowing them to culturally translate between the cooperating teams and bring them closer together.

Guideline 8: The choice of offshore partners should not only consider cost, but also technical limitations, cultural compatibility, and present expertise.

3.3.2 Virtual Teams

Virtual teams themselves already have major consequences for most of the development process. While the cooperation in a purely digital context introduces additional challenges to aspects such as teamwork and communication, the literature also includes distributed teams in general in this concept. Since distributing teams across different locations commonly leads to the formation of virtual teams because there are no real alternatives, both parts of the concept often share similar resulting challenges.

The increased complexity of working in virtual teams makes coordination more difficult for managers. Pare and Dube (1999) state that it might even result in a loss of control over the managed teams or reduce the impact managers can have on the development process. Herbsleb et al. (2001) describe the increased number of people involved in virtual teams as a reason for the consequent increase in needed coordination and communication meetings which then are signs of increased difficulty in coordination. According to Pare and Dube (1999), another reason could be that it is more difficult to keep track of team members' activities across multiple locations. Additionally, the reduction in informal communication is named as a relevant factor for estimation and scheduling errors in (Casey and Richardson 2006).

While Pare and Dube (1999) suggest that standardised methods can be used to prevent managers from losing the relevant overview, Herbsleb et al. (2001) promote the careful choice of correct tools to support the virtual process as a relevant approach to the increased coordination requirements. Tools that visualise the development process can help raise managers' awareness of tasks in progress, while also aiding in avoiding code control errors (Al-Ani et al. 2008). Finally, Jiménez et al. (2009) argue that reducing dependencies between distributed teams or its members can reduce the overall coordination difficulties.

Not only managers are negatively affected in their coordination by distributing teams. According to Jiménez et al. (2009), the team members themselves can be isolated when working purely digitally. As a result, they may struggle to be aware of the other team members' active tasks as well as the way knowledge is distributed between them, making it difficult to acquire key information for their own work.

Again, different visualisation tools that communicate this knowledge distribution or current processes are supported as approaches to solve these problems by a number of articles such as (Froehlich and Dourish 2004). Furthermore, thorough

documentation of roles and structures is mentioned in (Karolak 1999) as a method to increase clarity and awareness in distributed teams.

Poor communication can also have indirect effects on projects, as Stables (2001) describes how it can lower job satisfaction and increase employees' stress levels, while Hargie et al. (2002) name high employee turnover and lower commitment as consequences of lacking communication.

As already described, tools are a prevalent answer to problems in the virtual context. Jiménez et al. (2009) warn that these should always have to be carefully chosen with the context, project, and teams in mind. Additionally, with the introduction of distance, the reliance on internet connection increases, meaning that the correct choice of tooling can also include the consideration of available bandwidths. Additionally to the previously mentioned visualisation tools suggested in (Al-Ani et al. 2008), Bruegge et al. (2006) also propose a tool for collaborative code inspections to combat coding errors, which may have an increased impact in virtual development processes (Jiménez et al. 2009).

The distribution of teams can lead not only to an isolation of team members regarding coordination, but also regarding the team's cohesion and team feel (Holmstrom et al. 2006). Pare and Dube (1999) argue that this is caused by reduced social interactions in virtual teams and can result in a reduced commitment to the project and increase conflicts. According to Moe and Šmite (2008), reduced socialisation also causes a lack of trust between team members, which further reduces productivity.

Karolak (1999) describes the importance of a separate conflict resolution process to handle conflicts when face-to-face meetings are not possible. Pare and Dube (1999) also emphasise the value of specific rules for conflicts and suggest that meeting in person at the start of new virtual projects is highly beneficial to the team members.

Finally, decentralising the process regarding location by distributing teams across locations may not be compatible with decentralising other aspects such as organisational structure. Even though the scrum framework encourages smaller teams and self-management (decentralisation of hierarchy), Vallon et al. (2013) describe multiple incompatibilities when trying to enact scrum in virtual teams. Firstly, they argue that distribution reintroduces hierarchical management tendencies into the process, reversing the empowerment of the teams. Secondly, the increased difficulty to communicate and to coordinate is detrimental to the scrum process that encourages on team awareness. Consequently, they advise against

geographically distributing scrum teams and argue that the relevant scrum roles have to be present at each location if the distribution is unavoidable.

Guideline 9: Insofar possible, dependencies between distributed parts of a virtual team should be minimized. For all the remaining aspects of a distributed project, coordination and cooperation should be supported by facilitating transparency and effective communication and consequently increasing the project awareness of all team members.

Guideline 10: To facilitate proper team building in a virtual setting, team members should meet in person at the beginning of any distributed cooperation and if possible, the socialisation and communication should be supported by regular face-to-face meetings. In some cases, video conferencing can supplement the other socialisation efforts.

3.3.3 Home Office

Together with cooperating offshore partners, working from home is one of the most extreme forms of decentralising location. As it guarantees working virtually, even when the rest of team is not necessarily doing so, it always involves the same challenges as the other forms of distributed cooperation, while also introducing additional challenges on a more personal level.

Firstly, working from home often includes separating team members entirely from the rest of the team, which only increases the negative effects distribution can have on cooperation. Scott and Timmerman (1999) state that removing a team member from the others decreases the project awareness of that employee. Both (Marshall et al. 2007) and (Mann et al. 2000) argue that the lack of support, social interactions and feeling of group belonging increase workplace isolation and the resulting problems. Kurland and Cooper (2002) also describe that employees working from home feel like they are recognized significantly less for their achievements than their co-workers working in a co-located fashion.

According to (Jones 1997) home office often leads to a lack of separation between work and the private life of employees. Zhang (2016) describes that the blurring of that line can lead to employees being overworked and consequently experiencing burn out, because they have to face constant demands from both sides of their life. Additionally, they describe how conflicts of either side may influence the other because of the poor separation, meaning that a stressful private life has an increased negative effect on work performance. Bailey and Kurland (2002)

argue that employees working from home tend to work for more hours, which supports the risk for overworking.

Guideline 11: When home office is used, the separation between work and private life should be facilitated by encouraging the use of a separate workspace inside the employees' homes or informing about the importance of measures such as removing work equipment from the living space at the end of a workday.

3.4 Process of Decentralisation

The process of decentralisation in the context of this work means any of the above forms of decentralisation in motion and before completing the transition towards the decentralised system. This obviously comes with several challenges.

3.4.1 Customer Cooperation

As software development projects are often performed in a customer-provider relationship rather than purely in-house, the cooperation with such customers can be critical to the project's success. Decentralisation affects this cooperation in two major ways according to the literature, the increased importance of cooperation in decentralised contexts as well as the consequences of decentralisation on customer communication. Because distribution results in increased coordination and communication needs, the customer has to be available more frequently and has to be more cooperative.

Korkala et al. (2009) argue that facilitating customer cooperation is mostly dependent on positive relationships with the customers. Additionally, they state that the policies on the supplier's side must be compatible to the customer, to encourage cooperation. According to Bergadano et al. (2014), while decentralisation increases the need for cooperation, it also actively hinders it. As communication in general gets more difficult in virtual or distributed settings, customer cooperation also suffers in those cases and needs to be addressed in a similar fashion.

Guideline 12: Decentralised customer cooperation can be facilitated by encouraging communication, improving the relationship to the customer, and the use of bridgeheads.

3.4.2 Resistance to Change

The support of current leadership during significant changes can be crucial to the success of any form of

transition and is therefore also relevant for decentralisation. Resistance to change can occur in many forms and not only on a leadership level. Where resistance has significant impacts and where it is most common is discussed in this concept.

One of the previous concepts is most prominently met with resistance: The decentralisation of hierarchy that often includes the empowerment of teams. Strauss (1982) states that it is common for the people currently holding the power to actively resist against handing it further downwards. Even when the process has already progressed, resistance can still emerge within the now empowered team. According to (Barker 1993) informal differences in power can reintroduce themselves in self-managed teams, while (Gruenfeld and Tiedens 2010) and (Pfeffer 2013) argue that both formal and informal forms of hierarchy re-emerge because of personal drives for success and psychological processes which are also responsible for the endurance of hierarchy in the first place.

Argyris (1998) suggests, that internal commitment and personal psychological development are relevant factors against the difficulties of self-management and the trend towards hierarchy. They also describe that the defensive behaviour leading to these problems can be addressed by strengthening different values and mindsets.

Gruenfeld and Tiedens (2010) and Pfeffer (2013) describe the creation of a formal system and formal rules for the decentralisation as a critical success factor to overcoming the dominance of hierarchical organisation. This is supported by Adler et al. (1999) who argue that this formalisation can be helpful in communicating the consequences of the decentralised structures on daily work to newly integrated employees.

Guideline 13: Resistance to changes in hierarchy can be addressed by establishing formal rules and structures for the process as well as investing into shared values that align with the desired organisational structure.

3.4.3 Staffing Decentralised Teams

One major benefit attributed to distributing teams across sites and especially to cooperating in an offshore context, is the availability of staff. The wider range of available staff and lower prices induced by decentralisation can be a driving reason to transition to more decentralised approaches.

Krishna et al. (2004) warn that together with the cultural differences introduced by working across different countries, a certain difference in employee

motivation also emerges. For example, in Japan a comparatively higher salary might not be the most important factor in recruiting capable staff, as the social standing of the company may be of more relevance to the potential employees.

Guideline 14: Cultural differences should be considered when establishing the strategy to recruit valuable staff in offshore contexts.

3.5 Decentralisation in General

Various aspects of decentralisation are discussed in the literature that show that decentralisation is not the correct choice in certain cases. Incompatibility of certain employees to distributed work environments, the negative interaction between decentralisation of hierarchy and decentralisation of location as well as the need to recentralise knowledge in decentralised projects are all examples for situations in which decentralisation can best be facilitated by carefully

choosing which part of a given system to centralise. With the huge variety of projects, their individual requirements, and compatibilities that can influence the success of decentralisation, the most relevant success factor for decentralisation can often be the choice to decentralise only parts of software developments that are both suited for decentralisation and will benefit from it. Finally, this leads to the last and perhaps most important guideline.

Guideline 15: Regardless of form, decentralisation is not always the correct choice for every project. Because possible benefits and drawbacks are highly dependent on individual project contexts, the careful choice to decentralise a given part of a project should also be made on an individual basis. Forcing decentralisation on any part of a software development project should be avoided.

An overview of the fifteen guidelines developed in the course of this work is given in Table 1.

Table 1: Overview of the developed guidelines.

No.	Content
1	Employees should be assigned to only one team at a time whenever possible. Team building efforts should be performed to increase the benefits of team belonging for the team members that cooperate with each other in reality rather than in theory.
2	When managing project risk in multi-team systems, the impact of the approach should be considered as to not underestimate the real risk that is present.
3	For development in decentralised project settings, complement or even replace e-mail communication with better-suited communication tools such as tools that include chatroom functionalities or video conferencing while also paying attention to compatibility across the organisation as well as the necessary user training.
4	In decentralised projects, knowledge should be centralised using central documentation or a knowledge database including functionalities to effectively navigate and search for key information within it.
5	Knowledge should be decentralised to increase focus and avoid overloading individual team members with information. To make use of its benefits as well as avoid unawareness of where knowledge is located, communication tools, regular meetings and a bridgehead role that holds meta knowledge can be employed. The bridgehead role should be dedicated and "full time" to guarantee availability and effective knowledge distribution.
6	To facilitate offshore cooperation, measures such as team building efforts, cultural training and processes adjustments should be performed bilaterally in order to bring the cooperating teams closer together instead of attempting to adjust offshore teams to local standards.
7	In cooperation involving different cultures, the bridgehead role gains importance and should be filled with a team member that has experience in working in and with the partners culture, allowing them to culturally translate between the cooperating teams and bring them closer together.
8	The choice of offshore partners should not only consider cost, but also technical limitations, cultural compatibility, and present expertise.
9	Where possible, dependencies between distributed parts of a virtual team should be minimized. For all the remaining aspects of a distributed project, coordination and cooperation should be supported by facilitating transparency and effective communication and consequently increasing the project awareness of all team members.
10	To facilitate proper team building in a virtual setting, team members should meet in person at the beginning of any distributed cooperation and if possible, the socialisation and communication should be supported by regular face-to-face meetings. In some cases, video conferencing can supplement the other socialisation efforts.
11	When homeoffice is used, the separation between work and private life should be facilitated by encouraging the use of a separate workspace inside the employees' homes or informing about the importance of measures such as removing work equipment from the living space at the end of a workday.

Table 1: Overview of the developed guidelines (cont.).

No.	Content
12	Decentralised customer cooperation can be facilitated by encouraging communication, improving the relationship to the customer, and the use of bridgeheads.
13	Resistance to changes in hierarchy can be addressed by establishing formal rules and structures for the process as well as investing into shared values that align with the desired organisational structure.
14	Cultural differences should be considered when establishing the strategy to recruit valuable staff in offshore contexts.
15	Regardless of form, decentralisation is not always the correct choice for every project. Because possible benefits and drawbacks are highly dependent on individual project contexts, the careful choice to decentralise a given part of a project should also be made on an individual basis. Forcing decentralisation on any part of a software development project should be avoided.

4 CONCLUSION

In this work, the challenges decentralisation of different aspects of the software development introduces, were explored. To answer the posed research questions, a literature review covering articles about decentralisation has been conducted to establish existing challenges and possible approaches to address them and consequently facilitate the decentralisation of software development. The contents of the found literature were compared and combined into 15 guidelines meant to assist both in transitioning towards decentralisation as well as operating in existing decentralised projects and structures.

As it was shown, decentralisation can affect many different aspects of software development projects. It leads to various challenges that differ between forms of decentralisation and can be approached in a variety of ways. As such, there is no one definitive answer to the present research questions, but rather many answers of varying specificity and impact. To summarize: There are many challenges, yet they often involve either incompatibility of project aspects with or a lacking adjustment to the new decentralised requirements. While the approaches to them are numerous as well, recognising the challenges and acting upon the need for adjustments is often the most critical step. Lastly, the proposed guidelines can facilitate decentralisation, especially if they are applied with the individual requirements of a given project in mind.

The reviewed literature showed, that even without the COVID-19 pandemic enforcing it, decentralisation is very much present in many if not all software development endeavours of the time.

Regarding the scientific body of knowledge, the publication at hand can be most closely grouped together with the mentioned articles that address multiple challenges or aspects of decentralisation. It

aims to fill the gap in articles covering decentralisation as a whole by combining both coverage and detail, consequently reducing the degree of specificity of both results and challenges while still differentiating between forms of decentralisation. As a result of evaluating the existing literature relatively extensively, the present study is very problem-oriented and can only present more specific approaches to some of the uncovered challenges. Therefore, it can be grouped between the challenge-oriented literature and the articles covering very broad approaches.

In future studies the guidelines could be evaluated in real-life contexts. Applying them to various real development projects can be done to explore their validity and applicability, possibly even in long term studies. To further allow for generalisation, similar studies may be executed in very different companies regarding size and field.

REFERENCES

- Adler, Paul S. (2001): Market, Hierarchy, and Trust: The Knowledge Economy and the Future of Capitalism. In *Organization Science* 12 (2), pp. 215–234.
- Adler, Paul S.; Goldoftas, Barbara; Levine, David I. (1999): Flexibility Versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System. In *Organization Science* 10 (1), pp. 43–68.
- Adler, Paul S.; Kwon, Seok-Woo; Heckscher, Charles (2008): Perspective—Professional Work: The Emergence of Collaborative Community. In *Organization Science* 19 (2), pp. 359–376.
- Al-Ani, Ban; Trainer, Erik; Ripley, Roger; Sarma, Anita; van der Hoek, André; Redmiles, David (2008): Continuous coordination within the context of cooperative and human aspects of software engineering. In Li-Te Cheng (Ed.): Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering. New York, USA. ACM.

- Argyris, Chris (1998): Empowerment: The emperor's new clothes. In *Harvard business review* 76 (3), pp. 98–105.
- Aspray, William; Mayadas, Frank; Vardi, Moshe Y. (2006): Globalization and offshoring of software. In *Report of the ACM Job Migration Task Force, Association for Computing Machinery*.
- Babar, Muhammad Ali; Kitchenham, Barbara; Zhu, Liming; Gorton, Ian; Jeffery, Ross (2006): An empirical study of groupware support for distributed software architecture evaluation process. In *Journal of Systems and Software* 79 (7), pp. 912–925.
- Bailey, Diane E.; Kurland, Nancy B. (2002): A review of telework research: findings, new directions, and lessons for the study of modern work. In *Journal of Organizational Behavior* 23 (4), pp. 383–400.
- Barker, James R. (1993): Tightening the Iron Cage: Conservative Control in Self-Managing Teams. In *Administrative Science Quarterly* 38 (3), p. 408.
- Behr, Terry A.; Jex, Steve M.; Stacy, Beth A.; Murray, Marshall A. (2000): Work stressors and coworker support as predictors of individual strain and job performance. In *J. Organiz. Behav.* 21 (4), pp. 391–405.
- Bergadano, Francesco; Bosio, Gianni; Spagnolo, Stefano (2014): Supporting Collaboration between Customers and Developers. In *International Journal of Distributed Systems and Technologies* 5 (2), pp. 1–16.
- Bernstein, Ethan; Bunch, John; Canner, Niko; Lee, Michael (2016): Beyond the holacracy hype. In *Harvard business review* 94 (7), p. 8.
- Brannen, Mary Yoko; Salk, Jane E. (2000): Partnering Across Borders: Negotiating Organizational Culture in a German-Japanese Joint Venture. In *Human Relations* 53 (4), pp. 451–487.
- Bruegge, Bernd; Dutoit, Allen; Wolf, Timo (2006): Sysiphus: Enabling informal collaboration in global software development. In : International Conference on Global Software Engineering 2006: IEEE.
- Carey, Jane M. (1998): Creating global software: A conspectus and review. In *Interacting with Computers* 9 (4), pp. 449–465.
- Casey, Valentine; Richardson, Ita (2006): Project Management within Virtual Software Teams. In : International Conference on Global Software Engineering 2006: IEEE.
- Cohen, Susan G.; Ledford, Gerald E. (1994): The Effectiveness of Self-Managing Teams: A Quasi-Experiment. In *Human Relations* 47 (1), pp. 13–43.
- Contreras, Françoise; Baykal, Elif; Abid, Ghulam (2020): E-Leadership and Teleworking in Times of COVID-19 and Beyond: What We Know and Where Do We Go. In *Frontiers in psychology* 11, p. 590271.
- Cooper, Harris M. (1988): Organizing knowledge syntheses: A taxonomy of literature reviews. In *Knowledge in Society* 1 (1), pp. 104–126.
- Cramton, Catherine Durnell (2001): The Mutual Knowledge Problem and Its Consequences for Dispersed Collaboration. In *Organization Science* 12 (3), pp. 346–371.
- Da Silva, Fabio Q.B.; Costa, Catarina; Franca, A. Cesar C.; Prikladinicki, Rafael (2010): Challenges and Solutions in Distributed Software Development Project Management: A Systematic Literature Review. In : Proceedings of the 5th IEEE International Conference on Global Software Engineering. Princeton, NJ, USA: IEEE, pp. 87–96.
- Daft, Richard L.; Lengel, Robert H. (1986): Organizational Information Requirements, Media Richness and Structural Design. In *Management Science* 32 (5), pp. 554–571.
- Forster, Nick (2000): Expatriates and the impact of cross-cultural training. In *Human Res Manag J* 10 (3), pp. 63–78.
- Froehlich, J.; Dourish, P. (2004): Unifying artifacts and activities in a visual tool for distributed software development teams. In : Proceedings, 26th International Conf. on Software Engineering. ACM. New York, N.Y.
- Gruenfeld, Deborah H.; Tiedens, Larissa Z. (2010): Handbook of social psychology. 5th. ed. Hoboken, NJ: John Wiley & Sons Inc.
- Hargie, O.; Tourish, D.; Wilson, N. (2002): Communication Audits and the Effects of Increased Information: A Follow-up Study. In *Journal of Business Communication* 39 (4), pp. 414–436.
- Hellström, Tomas; Malmquist, Ulf; Mikaelsson, Jon (2001): Decentralizing knowledge: managing knowledge work in a software engineering firm. In *The Journal of High Technology Management Research* 12 (1), pp. 25–38.
- Herbsleb, J. D.; Mockus, A.; Finholt, T. A.; Grinter, R. E. (2001): An empirical study of global software development: distance and speed. In Hausi A. Müller (Ed.): Proceedings of the 23rd International Conference on Software Engineering. ACM SIG Software Engineering: IEEE Computer Society.
- Holmstrom, Helena; Conchuir, Eoin; Agerfalk, Par; Fitzgerald, Brian (2006): Global Software Development Challenges: A Case Study on Temporal, Geographical and Socio-Cultural Distance. In : International Conference on Global Software Engineering 2006: IEEE.
- Jamous, Naoum; Garttan, Gaurav; Staegemann, Daniel; Volk, Matthias (2021): Hybrid Project Management Methods Efficiency in IT Projects. In : Proceedings of the AMCIS 2021. Montreal, Canada.
- Jiménez, Miguel; Piattini, Mario; Vizcaino, Aurora (2009): Challenges and Improvements in Distributed Software Development: A Systematic Review. In *Advances in Software Engineering* 2009, pp. 1–14.
- Jones, Marian M. (1997): Out of the office, out of control. In *Psychology Today*, 30(2), 16.
- Karolak, Dale Walter (1999): Global software development: managing virtual teams and environments: IEEE Computer Society Press.
- Kilduff, Gavin J.; Elfenbein, Hillary Anger; Staw, Barry M. (2010): The Psychology of Rivalry: A Relationally Dependent Analysis of Competition. In *AMJ* 53 (5), pp. 943–969.
- Korkala, Mikko; Pikkarainen, Minna; Conboy, Kieran (2009): Distributed Agile Development: A Case Study of Customer Communication Challenges. In P.

- Abrahamsson, M. Marchesi, F. Maurer (Eds.): Agile Processes in Software Engineering and Extreme Programming. XP 2009: Springer (LNBIP, 31), pp. 161–167.
- Krishna, S.; Sahay, Sundee; Walsham, Geoff (2004): Managing cross-cultural issues in global software outsourcing. In *Commun. ACM* 47 (4), pp. 62–66.
- Kurland, Nancy B.; Cooper, Cecily D. (2002): Manager control and employee isolation in telecommuting environments. In *The Journal of High Technology Management Research* 13 (1), pp. 107–126.
- Lanaj, Klodiana; Hollenbeck, John R.; Ilgen, Daniel R.; Barnes, Christopher M.; Harmon, Stephen J. (2013): The Double-Edged Sword of Decentralized Planning in Multiteam Systems. In *AMJ* 56 (3), pp. 735–757.
- Leavitt, Harold J. (2005): Top down: Why hierarchies are here to stay and how to manage them more effectively: Harvard Business Press.
- Lee, Michael Y.; Edmondson, Amy C. (2017): Self-managing organizations: Exploring the limits of less-hierarchical organizing. In *Research in Organizational Behavior* 37, pp. 35–58.
- Levy, Yair; Ellis, Timothy J. (2006): A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. In *InformingSciJ* 9, pp. 181–212.
- Lundberg, Craig C.; Thompson, James D. (1967): Organizations in Action. In *Administrative Science Quarterly* 12 (2), p. 339.
- Magee, Joe C.; Galinsky, Adam D. (2008): Social Hierarchy: The Self-Reinforcing Nature of Power and Status. In *ANNALS* 2 (1), pp. 351–398.
- Mäki, Eerikki; Järvenpää Eila; Ziegler, Kirsi (2004): Communication and Knowledge Sharing in a Decentralized Organization.
- Mann, Sandi; Varey, Richard; Button, Wendy (2000): An exploration of the emotional impact of tele-working via computer-mediated communication. In *Journal of Managerial Psych* 15 (7), pp. 668–690.
- Marshall, Greg W.; Michaels, Charles E.; Mulki, Jay P. (2007): Workplace isolation: Exploring the construct and its measurement. In *Psychology & Marketing* 24 (3), pp. 195–223.
- Maznevski, Martha L.; Chudoba, Katherine M. (2000): Bridging Space Over Time: Global Virtual Team Dynamics and Effectiveness. In *Organization Science* 11 (5), pp. 473–492.
- Moe, Nils Brede; Šmite, Darja (2008): Understanding a lack of trust in Global Software Teams: a multiple-case study. In *Software Process: Improvement and Practice* 13 (3), pp. 217–231.
- Mohan, Kannan; Ramesh, Balasubramaniam (2007): Traceability-based knowledge integration in group decision and negotiation activities. In *Decision Support Systems* 43 (3), pp. 968–989.
- Paasivaara, Maria; Durasiewicz, Sandra; Lassenius, Casper (2009): Using Scrum in Distributed Agile Development: A Multiple Case Study. In : 4th IEEE International Conference Global Software Engineering. IEEE. Piscataway, NJ: IEEE.
- Pakos, Oscar; Walter, Jana; Rücker, Marc; Voigt, Kai-Ingo (2021): The Leap into the New Normal in Creative Work: A Qualitative Study of the Impact of COVID-19 on Work Practices in Industrial Companies. In *EJBM* 13 (10).
- Pare, Guy; Dube, Line (1999): Virtual Teams: An Exploratory Study of Key Challenges and Strategies. In : Proceedings of the 20th ICIS. Charlotte, North Carolina, USA.
- Pfeffer, Jeffrey (2013): You're Still the Same: Why Theories of Power Hold over Time and Across Contexts. In *AMP* 27 (4), pp. 269–280.
- Pinsonneault, Alain; Boisvert, Martin (2001): The impacts of telecommuting on organizations and individuals: A review of the literature. In *Telecommuting and virtual offices: Issues and opportunities*, pp. 163–185.
- Sarkar, Santonu; Sindhgatta, Renuka; Pooloth, Krishnakumar (2008): A collaborative platform for application knowledge management in software maintenance projects. In : 1st Bangalore Annual Compute Conference. New York, USA: ACM.
- Scott, C. R.; Timmerman, C. E. (1999): Communication technology use and multiple workplace identifications among organizational teleworkers with varied degrees of virtuality. In *IEEE Trans. Profess. Commun.* 42 (4), pp. 240–260.
- Staples, D. Sandy (2001): A Study of Remote Workers and Their Differences from Non-Remote Workers. In *Journal of Organizational and End User Computing* 13 (2), pp. 3–14.
- Strauss, George (1982): Worker participation in management — An International perspective. In *Research in Organizational Behavior* 4, p. 173.
- Tajfel, Henri (1982): Social Psychology of Intergroup Relations. In *Annu. Rev. Psychol.* 33 (1), pp. 1–39.
- Turnbull, Darren; Chugh, Ritesh; Luck, Jo (2021): The Use of Case Study Design in Learning Management System Research: A Label of Convenience? In *International Journal of Qualitative Methods* 20, 160940692110041.
- Tversky, Amos; Kahneman, Daniel (1983): Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. In *Psychological review* 90 (4), p. 293.
- Vallon, Raoul; Strobl, Stefan; Bernhart, Mario; Grechenig, Thomas (2013): Inter-organizational Co-development with Scrum: Experiences and Lessons Learned from a Distributed Corporate Development Environment. In H. Baumeister, B. Weber (Eds.): Agile Processes in Software Engineering and Extreme Programming. XP 2013: Springer (LNBIP, 149), pp. 150–164.
- vom Brocke, Jan; Simons, Alexander; Niehaves, Björn; Reimer, Kai; Plattfaut, Ralf; and Cleven, Anne (2009): Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. In *ECIS 2009 Proceedings*.
- Zhang, Jindan (2016): The Dark Side of Virtual Office and Job Satisfaction. In *IJBM* 11 (2), p. 40.
- Zhugue, Hai (2002): Knowledge flow management for distributed team software development. In *Knowledge-Based Systems* 15 (8), pp. 465–471.