

# Patterns for Distributed Teams Revisited

LISE B. HVATUM

---

This paper is a rewrite of a set of patterns for distributed software development teams initially documented between 2004 and 2008. This way of working has become much more common today, and the current pandemic situation with many software professionals working from home has further accelerated the number of people who are working in a distributed context. The goal for this paper is to present the patterns in a shorter more accessible form, and with a better cohesiveness across the collection. A couple of short stories are used to illustrate the use of the patterns in sequences to reach a larger goal, and finally there are some pointers to other work on distributed teams.

Categories and Subject Descriptors: TBD

General Terms: Software Management, Distributed Teams

Additional Key Words and Phrases: TBD

## ACM Reference Format:

Hvatum, L. 2020. Patterns for Distributed Teams Revisited. 27<sup>th</sup> Conference on Pattern Languages of Programming (PLOP), PLOP 2020, Oct ??- ?? 2020, x pages.

---

## 1. INTRODUCTION

Several years ago I published some papers containing patterns for distributed teams. These were capturing experience from people who had worked on software development teams with members distributed geographically between Asia and the US. Since then, I have had more exposure to this way of working, to the point where it has become the norm rather than the exception. The initial work was based on experience from an inter-company context. I now have experience both with larger inter-company projects with a more complex geographical distribution, and with outsourcing which brings another layer of complexity. I have always wanted to create a more cohesive compilation of my initial work on distributed teams, and with PLOP 2020 being a virtual conference it felt like the time is right to do this. I am taking the opportunity to improve the patterns by incorporating additional learning and the opportunities brought by better tooling and agile and DevOps software practices, but I am also drastically scaling down the length of each pattern. The idea is that the patterns will live in two forms – a short form in this paper, and a longer form online in an ecosystem that can also include stories, examples, alternative pattern sequences, and discussions. I will discuss this part more in the section of Future Work.

## 2. BACKGROUND

The initial pattern collection was developed between 2004 and 2008, and all patterns papers were workshopped at the PLOP or EuroPLOP conferences. The work was at first a collaboration with a few colleagues, but was eventually taken over by me. I will credit the original authors at the end of this paper. For this work I am rewriting all the patterns. The form and style was different due to the variety of authors. This made them less coherent as a collection. I have also learned so much more about pattern writing since then. And finally I want to use a much shorter form for this paper to more quickly convey the essence of the practice and make it easier for the reader to get the overview.

Due to confidentiality reasons, I will only give a generic description of the patterns context. The original patterns were developed to capture the learnings from key personnel that had successfully developed products with some of the staff at a technology center in Asia, and the remaining staff at a technology center in the US. Traveling was possible, and the team members were able to meet occasionally. The staff at the location in Asia were less experienced, and had less business domain expertise at the time. This is no longer

the case. Product development within this organization today is distributed on several locations across the world with key staff being anywhere in this distribution. Traveling was reduced due to financial and practical restrictions but was still happening until the 2020 pandemic hit. Many of the team members know each other well from earlier assignments, and staff is regularly transferring between the locations.

Tooling has developed significantly since 2004. These patterns are more about human interaction than about tooling, but there will be an impact (explicitly in some cases, implicitly in other). (more on tooling??)

In regards to outsourcing, this is not the primary focus of these patterns. The need for communication and team interaction is no different when running a development with external resources, but there are some contextual implications. I will address this later in the paper.

The first part of the paper is an introduction to the patterns collection. This is followed by the individual patterns, and then by patterns sequences to illustrate the strength of the patterns working together. Finally I will discuss topics like outsourcing, tooling, and future work.

### 3. THE DISTRIBUTED TEAMS PATTERNS

The distributed teams patterns collection currently has twenty individual patterns. These can be categorized based on scope into personal patterns (what an individual can apply), team patterns (that apply within a single team), project patterns (that apply for a project in an organizational context) and patterns for an organization or company (patterns that apply on a level above an individual project or program). This categorization is shown in figure 1. The reader will see that a few patterns are placed across the borders between categories, meaning that they apply in both to varying degree.

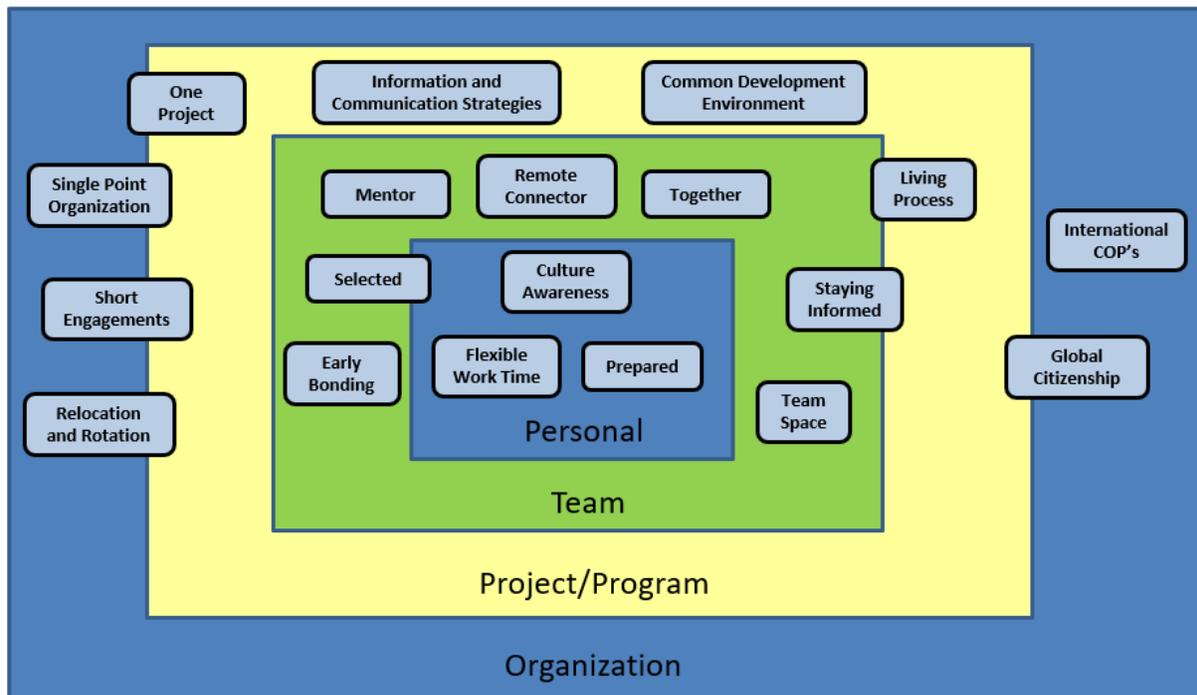


Figure 1: Patterns Collection

Although there are practices on the personal and team level, operating and organization with distributed teams require commitment and planning from all levels in the organization. Organizational practices around HR, finance and others need to be designed (or re-designed) to support the projects, and virtual teams need appropriate tooling and resources for their way of working.

One could easily argue that almost all of these patterns apply to any development team, and that is true. A practice like *early bonding* for example, applies to any team. But the forces and implementation are different in that it is harder and often more costly to do for distributed team, while the importance on the practice is stronger because of the remoteness of the team members.

A compressed version of the patterns giving the pattern name and purpose is presented in table 1:

Category	Pattern Name	Pattern Purpose
Personal	Prepared	The negative effect of not being co-located can be made significantly smaller by applying good practices from the start of the project, so make sure all are trained on distributed development from the start and are aware of the challenges.
Personal	Culture Awareness	Avoid collaboration problems that stem from a lack of understanding of ethical and behavioral code by training all team members in cultural aspects covering the various locations.
Personal	Flexible Work Time	Reduce the impact of working early/late to communicate with distant team members by being flexible on the mandatory work hours.
Team	Selected	Working on a distributed team is demanding and requires flexibility and good social skills so carefully select individuals and build the team gradually over time for maximum cohesion.
Team	Early Bonding	Trust and personal connections need to be built early and are key to have a functional team.
Team	Staying Informed	By allocating common time and making team members available to each other at agreed times during the week, you can mimic spontaneous contact (with a delay), and you ensure frequent communication in the team.
Team	Together	Even with Smart Meetings, communication suffer by not being co-located so make sure the team physically meets during development.
Team	Team Space	Local team rooms need to accommodate visiting team members and have tools for communication with remote team members.
Team	Mentor	There is much to learn from colleagues that have managed distributed teams before so assign a mentor to a new team.
Team	Remote Connector	Distributed teams struggle with keeping on the same page between locations so designate a team member at each site to manage the flow of information and facilitate problem solving.
Project/ Program	One Project	To avoid conflicts between local goals and what is expected as product deliveries create a clear project identity across the locations and make sure all team members share the same team priorities.
Project/ Program	Living Process	Spell out fundamental values, methodology and techniques used and keep it up-to-date for all team members.
Project/ Program	Common Development Environment	Carefully identify, select and implement one single common development environment for the project team to use regardless of their location. All components of this common development environment must support multiple locations with different time zones, and come with worldwide 24/7 support.

Project/ Program	Information and Communication Strategies	To share information across the complete team set up an information structure for the global team that enable consolidated reporting of status and easy sharing of information.
Organization	Single Point Organization	Avoid conflicting objectives and unclear authority and responsibility by establishing a clear organization where decision-making authority lies with a single role at each level and for each function in the organization.
Organization	Relocation and Rotation	To build an international workforce with a strong common identity and enforce common work methods relocate a part of your workforce on a rotational basis to the various locations.
Organization	Short Engagements	Team Short-term assignments at the other location will enable team members to get to know other team members better.
Organization	Global Citizenship	To have a workforce where individuals are equally respected and feel equally valuable make each location comply with the company policies and give all employees the same opportunities and benefits.
Organization	International COPs	To stay competitive and benefit from the knowledge and creativity of your whole workforce grow and support technical communities of practice on the global level.

Table 1: Purpose of Distributed Teams Patterns

### 3.1 PERSONAL PATTERNS

The following three patterns focus on practices that mainly involve the individual team member.



## PREPARED

When you create an awareness in your team of the challenges that you may face working in a distributed fashion they will be less frustrated when problems arise, and better able to proactively avoid them in the first place. A bit of naiveté can be good in that the tasks do not seem overwhelming. But you do not want your team to run head first into any complex situation without preparing them for how to deal with it, especially not at a time when you need to focus on building the trust and relationships and good communication.

The challenges of working as a distributed team will vary with your actual situation and be different if people are working from home within close time zones or if they are far apart geographically. Other elements to be aware of is if team members already know each other, if they come from different backgrounds (and this has many facets – cultural, area of expertise, experience level, familiarity with the business domain, ...) if they have worked on distributed team before, and if they are new to the organization).

Preparing team members for working on a distributed team needs to cover both a practical and a mental level. Practical topics are to get familiar with the team workflows (*living process*) and tooling (*common development environment*), and to understand how the team members interact with each other (*together* and in *smart meetings*). The deeper and more difficult part is to prepare the team members for the psychological challenges of collaborating with people who may be far away both in time and space, how to build respect and trust and tight collaboration and to resolve conflicts (*culture awareness* and *early bonding*), and for the work style that is not a 9-5 situation but that require you to be available at times that work for the global team (*flexible work hours*) and where the social interaction in the team is mainly virtual. These preparations will be different if the team is getting establishes as the team will need to figure out about the needed preparations as they start and there will surely be a lot of learning and adjustments and maturing.

It may be that your organization has captured lessons learned and good practice for working as distributed teams. If not, there are plenty of resources online and even some good books [refs]. Or you may engage an experienced coach to help do training and support the team at the start of the distributed project if you have the resources to do so. Most important is to have an open conversation and to agree and implement practices that fit your team and your context, and to do this from the beginning. As part of these preparations you should include collaboration tools and workspace (both private and team level) to support your needs.

As the team matures, you can use retrospectives and other means (surveys, dedicated workshops) to reflect on how and how well the team functions and to decide on improvements. When new team members join they need to be *prepared* so they can join on equal footing with the other team members.



## CULTURE AWARENESS

Understanding of and respect for your colleagues is of course always important to create a foundation of trust within the team. If you have a diverse team it can be difficult for each team member to know what may be upsetting or awkward to others. Your company may have awareness training as an established practice (in some parts of the world there is legal liability for a company for lack of respect in the form of harassment), or you may need to create something specifically for your team.

As part of your training you can have team members present something about their own culture that they would like for the team to know. You can also bring up scenarios to discuss but in doing so you also need to execute awareness – some team members may have an issue with certain topics.

In addition to training you can establish a mechanism for bringing up problematic events and topics in an anonymous way – to a mail box or through the HR function or an ethics line. Not everyone is OK to bring up issues confronting other team members, but everyone need a way to communicate and be listened to.



## FLEXIBLE WORK TIME

If a team is distributed over a wide range of time zones the team members will end up working outside local work hours, possibly needing to be in meetings as early as 6am or as late as 10 pm. But working a regular day plus the required extra hours to allow same-time communication within the team will mean long days and short nights maybe several days per week.

In more traditional organizations where parts of the team are collocated in office locations, people may feel awkward about arriving late at the office, or leaving very early. Coming in late looks lazy. Leaving early is very visible. There is less visibility for those working from home in the evening or early morning. When the team is mostly distributed and many work full time from home this is probably less of an issue.

The key word here is flexibility, and it must go both ways. The organization must not hold team members of distributed teams to being present (in the office or online) during the full range of “normal” work hours, but rather (formally) establish a culture where team communication and performance (autonomy and trust) are valued.

While the organization is showing flexibility in work hours, the team members can agree to have defined “common time” where people do their best to be available. Common team sessions should be utilizing these time slots. Outside the time slots a team member can allocate their time in a flexible way. Working early/late and the pressure to be available will necessarily have an impact on the private life of the employee and his/her family. Flexible work hours should significantly reduce the stress with the opportunity to join an event at their child’s school, or shop during the day, or take care of other private things during what was traditionally work hours.

### 3.2 TEAM PATTERNS

The following eight patterns focus on practices at the level of the distributed team.



#### SELECTED

When setting up a new distributed team, and especially if this is not the common way of working in the organization, there should be some deliberations in regards to the profile of the team members. Depending on the level of distribution (time zone, diversity, number of remote members) one needs to consider not only skills but also personalities. If you have a choice between a Slack evangelist and avid blogger and someone who prefers to stay in a separate office with the door closed, you will probably have a more engaged team member in the first.

So while technical skills are important, there are personal traits that also should be considered: openness to change, curiosity about other cultures, people who like to travel, oral communication skills. These traits are not something you can test for directly, but a good interview designed to understand the personality of the candidate should at least indicate if you are hiring a person who fits in your team.

Over time, the members on a distributed team may not be a result of deliberate selection from the organization, but rather a form of natural selection as team members who are not comfortable with the way of working chose to leave the team.



#### EARLY BONDING

In a distributed team you are very likely to have a very heterogeneous group of people who need to build trust and good communication. Emphasizing on building the team from the start of the project will help you create a well-functioning team sooner than later. If at all possible the team should spend some time face to face, including social events (although you need to apply the *culture awareness* when planning these). Even if the team members cannot meet in person you can organize virtual events (ref?).

The interactions are not only about social interaction though. You want the team members to be aware of each other's strengths and skills, and to make sure that knowledge sharing and mentoring is happening whether formal or informal. And not least, the team needs to define and develop their collaborative workflows, for example selecting and adapting an agile methodology and tooling to their team specific situation.

The first few weeks on a new team may be stressful to the team members trying to sort out their roles and relationships. It is important to give the team time to figure this out before there is any commitments to produce with shippable quality. You should probably be open to replacing team members who opt out of the commitment to working with remote colleagues and/or are uncomfortable with the *flexible work time*.



## STAYING INFORMED

Good communication is important for any development team, to ensure the team has clarity on anything from the product being developed to design decisions to work distribution. Without communication there will be mistakes and misconceptions that over time negatively affect the product as well as the team performance. Co-located team members may engage face to face many times per day, but for remote teams that is more challenging especially if the team members are distributed over time zones with little overlap in regular work time.

The team can counter this by devising a number of ways to support frequent and spontaneous communication. Short meetings can be pre-scheduled and recurring to make sure people can plan for attendance. These meetings can have a combination of pre-defined focus (design, testing etc.) and open agendas to make sure certain topics are discussed often while still leaving time for free discussions and general socializing. Timing must be respectful of peoples personal time, and should distribute the “inconvenience” factor evenly among the team. They should be set within the established common time as part of the *flexible work time* concept.

In distributed teams, communication is often a mix of synchronous and asynchronous as questions are being raised and people chime in when they are available. Tooling like Slack can support spontaneous communications where the conversation is kept and additional team members can join in at a later time, and still know what was already discussed. This can turn into a de facto asynchronous meeting.



## TOGETHER

Virtual teams have generally proven to be viable, and many do successfully deliver products of high value and high quality. But there is no hiding that this way of working is challenging, and that small misconceptions and miscommunications can lead to design mistakes and product imperfections. If at all possible it is a good idea to bring the team physically together to build personal bonds, socialize, work together on-site, and brainstorm, plan, test and in general interact and discuss.

There is of course costs involved, both for the organization to support travel and hotels, and for the team members who will spend time away from home. Some of this cost can be reduced by having the team members visit each other’s locations so that a part of the team is at home. The drawback is that the team members who are home engage less.

This is a good opportunity to share the vision of the product under development, and do joint planning especially where there are significant dependencies across locations. It is also a great opportunity to celebrate team successes.

For teams doing time-boxed iterative development (for example Scrum), it can be a good idea to synchronize the end of a sprint and the start of the next with the face to face event. If the team meets every

3 months with 2 week sprints, they may run every 8<sup>th</sup> closing of one sprint and planning of the next sprint at this time. This gives insight in the shared planning process and an excellent opportunity to synchronize designs. The teams may also use this time to retrospect on recent development like completion of major new features.



## TEAM SPACE

For teams that work from office locations and that spend some time *together* it can make sense to create flexible work spots and use furniture that is light and flexible so that the space can be set up to support visiting team members. Meeting off site is not only more expensive, but it takes away some of the benefits. You want the “home team” to feel like good hosts and the visitors to experience the regular work days and shared space with information radiators and personal touches. And you want the local and visiting team members to mingle as much as possible and be able to do daily work together. Borrowing office space elsewhere in the organization cannot support this.

A different team space is the virtual space the team is sharing meaning where they store and share information. When selecting and configuring these digital tools you again want to pay attention to your needs and goals for collaboration, making it easy for team members to interact and to find information even when their remote colleagues who produced it are asleep. In tools like Slack, team communication (including history) is accessible to the whole team which is great for joining an ongoing discussion and being able to get up to speed quickly. Other tools that support this mindset if used appropriately are requirements/issue tracking tools like JIRA, where the discussion around an item stays with the item and is persisted.



## MENTOR

There is much to learn from colleagues who have been part of distributed teams already, from practical topics like setting up the *team space* and the *common development environment*, to helping on team practices and interaction. The mentoring may happen on several levels. It may be that a new member is joining a distributed team and you assign an experienced team member as a personal mentor. Or a developer with prior experience of distributed development is joining a new team to and is helping the whole team. Or someone in the organization outside the team could take on the role as mentor, possibly for more than one team as a shared resource (a coach). Or you may go as far as hiring a professional coach who could be helping implement agile practices in a distributed setting.



## REMOTE CONNECTOR

In a context where a team is distributed on a set of office locations with a number of team members in each location the communication between the co-located team members will normally be better than the communication flow between the locations. This may lead to problems in disseminating information that is important to the development. It may be that a new business request lead to priority changes but not all locations are aware and so they keep working based on the previous plans. Or architectural changes are not being communicated clearly to all who need to know. Sometimes you may not know who needs to know in other locations because you do not have the full view of what they are working on. This is of course more challenging on large programs than for small project teams.

By assigning a role in each location that has an explicit responsibility to communicate with the other locations can mitigate these problems. This role is the go-to for any information that should be shared within the team, but is also proactively working on knowledge sharing across the full team. The tooling used by the team play an important factor in this as well. The *Connector* role can both use and promote good tooling and how it is used – for instance making sure that discussions around a requirement happens inside the backlog tool and not in separate e-mails between some of the team members.

Sometimes this role gets filled in an organic way but if you are deliberately looking for a candidate you will want the person who has a good network preferably in more than one location, and who knows a lot about what is going on in the team. Preferably a senior person but not a manager if people hesitate to “bother” this person.

### 3.3 PROJECT AND PROGRAM PATTERNS

The following four patterns focus on practices at the level of the projects and programs. There may be multiple teams within a project or a program, or it may be managed as one large team.



## ONE PROJECT

A project where the team is distributed may deal with significant organizational challenges. We have seen several examples of distributed teams where the team members report directly to local management, where tooling and work practices are location specific, and where recognition and benefits and managed very differently from one site to another. This leaves the project as a loosely coupled organization where it is hard to share common goals and priorities, where team workflows and practices are hard to align, and where team members are not committed to the project but rather to their location. If the project is distributed internationally the challenges are usually more complex, not least for the financial and legal management.

With some flexibility in the organization it is absolutely possible to organize the project as the primary entity. It requires that HR can deal with direct reporting lines across countries, that accounting can develop a model where a project budget is allocated and monitored on several locations, and where a project has one manager, a dedicated team, and autonomy to define their workflows and select their tooling to be the same across the full project. Project priorities apply to the whole team, and shared successes are celebrated and full credit is given to all team members.



## LIVING PROCESS

Even if the team is using a well-documented process like Scrum there are always implementation details that need to be known to all the team members. This can be details about how you manage the backlog, your particular “definition of Done”, where you keep different types of artifacts and how you share information. As you team and the product mature the process details will mature with it.

The help in the sharing and to make sure the current process knowledge is available to all team members at any time you will want to document it in a light way fashion. A wiki can be a good solution, as it is easy to use and still can keep the history of changes. The contents should be modifiable by any team member and seen as a shared responsibility. Depending on the nature of the product it may be required to document the team process (for instance for safety-critical systems, financial systems, medical systems etc.).

With remote team members the need to be explicit and to provide the process understanding and agreements in a document location is greater than when the team is co-located and more easily communicate the updates and tweaks to the process. It is also a greater chance that changes in the team practices is agreed between a subset of the full team, and so does not reach everyone unless there are established practices to update and broadcast process changes to everyone.



## COMMON DEVELOPMENT ENVIRONMENT

This may be less of an issue today than it was a few years ago as ALM (application lifecycle management) tools now commonly are deployed in the cloud and easy to share. Even then there should be attention to how the tools are used to manage code, requirements, and tests to not create artificial silos and obstacles in the development and deployment processes. In the *One Project* you want your *Living Process* to be driven by the business needs and the product architecture, not by the location of your team members. You want a single product backlog digitally managed and with full traceability across the backlog items to make the dependency management between the remote locations easy and transparent.



## INFORMATION AND COMMUNICATION STRATEGIES

Although a lot of the development information is shared in the ALM tool used for the development, there are additional content that more likely will be stored in other digital tooling. With a larger distributed team one may end up with a rather unstructured storage of key artifacts and project information if it is left to each individual to figure out where to store it. We all know how hard it can be to find an older file in our own document folder structure, and in content systems shared by many users this is just getting even harder. Usually there are alternative systems available – do you store in SharePoint or put it on the wiki, or should it maybe be in a controlled system managed by your legal or financial functions?

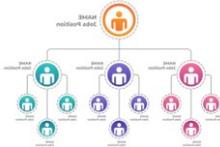
In our current “software climate”, and especially if delivering software solutions to a global market, the product is more than the code itself. You will need to make sure you manage 3<sup>rd</sup> part licenses, that you can document your cyber-security compliance, that you do not violate intellectual property or data right of use, that your user documentation is current and correct, etc. and there may be severe legal implications if you are unable to prove by documentation in case of accidents, lawsuits, contract violations etc.

It is a good idea to establish a clear information strategy and a shared information infrastructure for the *One Project*, and possibly assign a team member some responsibility to maintain this and to make sure the strategy is being followed.

Your communication strategy tells you who to inform, what to share, and the means of communication. This is not so much about your internal communication but about how you engage with your stakeholders and with other functions within your organization, and not least with users and customers. It is the complement of your information strategy – the information is the “what” and the communication is the “how” to share.

### 3.4 ORGANIZATION PATTERNS

The following five patterns focus on practices at the level of the overall organization in which the project or program lives. These patterns focus on how the organization can support the distributed teams and create a organizational foundation that is beneficial for this way of working.



#### SINGLE POINT ORGANIZATION

Establishing the *One Project* makes life easier for the team members who now have a clear and simple structure with clear leadership. But for the project management the situation will still be complex if they have to deal with a number of local organizational entities for HR, financial management, training, process and whatever other functions that are in place in the various locations. It may be difficult to get clarity on the budget and the spending, people's benefits and promotions and bonuses may be very different based on what is the norm in the country they work meaning team members are not treated equally, and local norms and standards may be in conflict with the project decisions.

With the proper attention in the organization there is a lot that can be done to smooth the way for the project. All functions (legal, HR, supply chain, HSE, ...) can identify one primary role that works directly with the project management and who is coordinating between the local functions to come up with solutions to handle various needs in a uniform way. For the handling of personnel it will normally not be possible or even the best solution to treat each team member exactly the same, but rather to implement fair systems that take into account the location and country norms as well as laws and regulations.

The location where the leader(s) in the organization are located often get a higher status than the other locations. To mitigate this one can distribute the leaders for the functions to create more balance, and the top leader can be purposely seated in a location that historically had less status and/or the role of the leader can be moved between locations when the current leader leaves the team. In organizations with several distributed teams, all leaders should not sit in the same location.



#### RELOCATION AND ROTATION

Within an organization, moving people between locations is a way to increase knowledge sharing and to create trust and friendship between people. This can happen on the virtual level, where people move between distributed teams, or it may involve people physically moving to other locations. Physical moves are of course harder to deal with as it has implications for the whole family and there is a substantial cost involved especially for international moves. But it is also the one that makes the individual more immersed in the culture of the new location, and that has a bigger learning potential because a person will gain understanding of the culture and ways of thinking, which again can help the team communication and workflows. The long term goal is to create a global workforce where you have a good network of people and were



## SHORT ENGAGEMENTS

Rather than a full physical relocation of a person to a new location it can be a great solution to organize short engagements where the team member spends a few weeks at another location. This could be a single team member working as part of the local team, or it could be that several team members come together in one location for a special period, for example to prepare for a new major release of the product. The nature of these engagements will be somewhat different. The single team member will become a normal part of the local team, while a group of team members from other locations will more likely influence the normal team dynamics. A challenge with these engagements when done on an international level is of course work permits and tax implication that the organization will have to resolve.



## GLOBAL CITIZENSHIP

It is a challenge to have team members from different countries, from cultures that may not be compatible, with differences in communication styles, with some having a better command of English (which is normally the shared language), and with gender and other types of identities. It may be interesting enough to make a local team function when one team member is a Chinese fashionista, one a Black Metal music fan, one an American fishing enthusiast, one a soccer fan, and one an Egyptian princess, with an age gap of 30 years between the youngest and the oldest. As a virtual team, how do you get these people to bond?

The organization has a lot of impact in the way it communicates and establishes the expectation of respect and equality across the organization, from company standards and policies to the practical day to day operation. By being strong and clear on the message of respecting and being professional towards your colleagues and by taking action early when there are problems, over time there will be an emerging culture of broad acceptance and collaboration between employees. But it requires a deliberate, focused and organized effort at every level of the organization, and it needs to show through promotions and assignments that what matters is your work achievements and not your skin color or your taste in music.



## INTERNATIONAL COPs

If your organization has enough software professionals you may want to consider establishing communities of practice (COPs) around core technologies and other topics. For this to work the interest and drive must come from the community, while the organization contributes by supporting the COP effort with tooling and time for COP leaders, funding for COP events and communication etc.

## 4. STORIES AND PATTERN SEQUENCES

Patterns are more powerful when chained together in sequences. This way they can reach bigger goals, and if applying one pattern leaves you with unwanted side effects another pattern can mitigate this and bring you further towards your desired outcome (ref. to Allan's book). An organization that is on a journey to seriously supporting distributed work probably work on several fronts – preparing the organizations structure and resources, while running pilot teams to gain a deeper understanding and to use this learning to guide the next steps in the process.

In this chapter, two stories are illustrating possible pattern sequences. The first is focusing on the team, while the second is focusing on the overall organization.

### 4.1 STORY A

In the early 2000, a technology company with a large presence in the US was setting up a new software center in Asia to be better positioned to sell their products in the Asian market. This was seen as a long term investment, and local resources were hired to build up the new center. A few key people from the US location transferred to the new center, but the company was left with a huge challenge to be able to deliver products that were jointly developed by the two locations. Most of the product and business knowledge was based in the US, and this was also true for the experience with development tools and processes.

The initial focus was on building the teams. Team members from both locations were carefully *Selected* for their language skills, for their ability to collaborate (people who were known to be inclusive, social, and good team players). Funding was made available for the teams members to visit each other for *Early Bonding*. This was also an opportunity for the new team members in Asia to learn about the business during their visit to the US location, and soon the teams were working on new products.

In the beginning, things did not go so smoothly. The project organization was informal and not so well planned. Although developers in Asia were assigned to teams in the US, their main reporting was to local managers. This caused confusion for the teams and the individuals on project priorities, authority and team identity. Occasionally developers were assigned to more than one team which clearly did not help. And to further complicate the situation, the engineering centers had different agendas, resulting in different priorities and objectives for the individuals. As an example, the center in Asia was planning for CMM certification. This meant extensive training for developers. The project managers sitting in the US were repeatedly experiencing that their developers were not working on the project, but the CMM training activity was not coordinated with the project plans and delivery dates and came as a complete surprise to the projects.

It was eventually understood that each team needed to be organized as *One Project*, to make sure the teams could focus on one set of priorities, and each team adopted a *Common Development Environment* and maintained their own *Living Process* within the overall software governance model in the company.

The process to get to a working model to almost two years, with trying and failing and gradually finding a working model. As the company was establishing other distributed teams, team members from the initial team were asked to function as a *Mentor*, and a couple of team members did *Short Engagements* in other locations to support new teams as they were starting up.

### 4.2 STORY B

Looking back at the organization from Story A today, they are very comfortable with working with distributed teams, and it has become the norm rather than the exception. People who were involved in the early distributed work have become managers, and because the company has focused on *One Class of Citizenship* the management of distributed teams is evenly balanced between the locations. With increasing business in Asia it is no longer a situation where one location is much stronger than the others in business

knowledge, process ownership or any other domain. The deliberate *Relocation and Rotation* practices means people are transferring across the organization thereby disseminating knowledge and creating equality.

Global standardization on both *People Management* and business systems tooling has supported the development of a *Single Point Organization*, so that a project now has no problems with managing budgets and other logistics independent on how many locations the team members are based in.

With the pandemic in 2020, this organization has converted to a scenario where all software teams work from home. With better collaboration tooling, time *Together* had already become less frequent, but employees worked from an office location and were used to interact with others across the individual teams on a daily basis. Team performance never went down but has actually increased somewhat which shows that these teams were used to the remote collaboration and has the resources to keep working together in the “new normal”.

## 5. TOOLING

The patterns do make reference to some tooling. This is meant as examples and in every case there are several tools for collaboration, for information management (and document control), and software development support available. Few teams today do not use digital tooling for most of this, but in a distributed scenario this is a must. At the minimum, one should have:

- Code management (version control, build and deploy tooling)
- Backlog management (requirements, defects, tests)
- User documentation management (user manual, online help, training, release notes, ...)
- Team collaboration (chat and file sharing, ...)

## 6. OUTSOURCING

The context of this patterns collection is the support of distributed teams within an organization. The writing is on purpose kept as open-ended as possible, but in some cases the practices do need a supporting organization to fund and drive initiatives. That said, if your team includes individual contractors and/or people from a consulting agency, you should still find value in these patterns. You could even improve the collection by adding your own content if you are experienced with outsourcing!

## 7. FUTURE WORK

These patterns were derived from mining within one organization several years ago, but they were then updated to fit the scenarios of today. There are several other pattern collections with distributed teams patterns, and it is my hope that we can continue working on this and bring these patterns together in a cohesive whole.

## 8. ACKNOWLEDGEMENTS

I was so very fortunate to get Neil Harrison as my shepherd for this paper. Not only is he an amazing shepherd but we share an interest in patterns for distributed work. Our discussions went broader than this paper. I am very grateful for his constructive help to make the contents more accessible, and I am looking forward to our continued collaboration to combine this with the broader knowledge base on the topic. (workshop TBD). I am also grateful to me old colleagues who were involved in the initial mining and documentation (Vincent, Ying, Denis, Thierry, and Adrien).

## REFERENCES

1. “Organizational Patterns of Agile Software Development” by James O. Coplien and Neil B. Harrison, ISBN 0-13-146740-9, Pearson Prentice Hall 2005
2. “Using an Agile Software Process with Offshore Development” by Martin Fowler, <http://www.martinfowler.com/articles/agileOffshore.html>
3. “Global Software Development Handbook” by Raghvinder Sangwan, Matthew Bass, Neel Mullick, Daniel J. Paulish, and Juergen Kazmeier, ISBN 0-8493-9384-1, Auerbach Publications, 2007
4. “Can absence make a team grow stronger?” by A. Majchrzak, A Malhotra, J. Stamps and J. Lipnack, Harvard Business Review, May 2004
5. “Interaction Patterns of Agile Development” by Jens Coldewey, 2004
6. “Capable, Productive and Satisfied” by Paul Taylor, PLoP 1998.
7. “201 Principles of Software Development” by A.M. Davis, McGraw-Hill, 1995
8. “Culture Clash – Managing the Global High-Performance Team” by Thomas D. Zweifel, ISBN 1-59079-051-0, Swiss Consulting Group 2003
10. “Mastering Virtual Teams – Strategies, Tools, and Techniques that Succeed” by Deborah L. Duarte and Nancy Tennant Snyder, ISBN 0-7879-5589-2, Jossey-Bass 2001
11. “Trust within Global Virtual Teams” by Olivier Chavaren, ISBN 0-595-27577-X, iUniverse, Inc 2003
12. “Virtual Teams – Reaching across Space, Time, and Organizations with Technology” by Jessica Lipnack and Jeffrey Stamps, ISBN 0-471-16533-0, John Wiley & Sons 1997
13. “The Manager’s Pocket Guide to Virtual Teams” by Richard Bellingham, ISBN 0-87425-615-1, HRD Press, Inc 2001
14. “Global Software Development – Managing Virtual teams and Environments” by Dale Walter Karolak, ISBN 0-8186-8701-0, IEEE Computer Society 1998
15. “Managing Virtual Teams – Practical Techniques for High-Technology Project Managers” by Martha Haywood, ISBN 0-89006-913-1, Artech House 1998
16. “Working Virtually – Managing People for Successful Virtual Teams and Organizations” by Trina Hoefling, ISBN 1-57922-032-0, Stylus Publishing 2001
17. The Distance Manager – A Hands-on Guide to Managing Off-Site Employees and Virtual Teams” by Kimball Fisher and Mareen Duncan Fisher, ISBN 0-07-13065-4, McGraw-Hill 2001