Patterns to Enable Distributed Development

Lise B. Hvatum

Sugar Land Technology Center, Houston, USA

Over the last 3 years, I have been part of a group capturing best practices for distributed software development within my company, an international service company with engineering centers in Europe, North America and Asia. Geographically distributed development teams have been a reality in the organization for decades between the European and American locations. Trends the last years with a build-up of involvement in Asia have increased the challenges of distributed teams, and made us aware of the need for establishing methodologies and practices in the area.

From a combination of interviews, workshops, written reports, and personal experience, we identified several practices that the teams applied as they matured in their capability to work across locations. Some of these were presented at previous PLoP conferences [EuroPLoP 2004, PLoP 2004, EuroPLoP 2005].

Presenting and discussing our patterns outside the company gradually made it clear to us how much implicit knowledge and company culture impact not only the understanding of the patterns, but also their applicability in an organization. We needed to take a step back and look at the organizational context in which our patterns can apply. Or rather, better understand what kind of organization can succeed in doing distributed development.

Fundamental to an organization is its underlying value system, and how this is instilled in the individuals. Our recognition of patterns for distributed development started on the tools and techniques level, but we soon realized that this was too simplistic. What is clearly most important to succeed are the people, their knowledge and skills, and even more important what they believe in:

"People are the key to success. Highly skilled people with appropriate skilled experience, talent and training are key. The right people with insufficient tools, languages and process will succeed. The wrong people with appropriate tools, languages and process will probably fail."
- From Davis' top 30 principles

The fundamental question we are asking in this paper is how an organization can succeed in doing distributed product development, as captured in the overall problem statement:

How do you grow an organization that smoothly, efficiently and effectively delivers successfully completed software projects of high quality independently of the physical locations of the individuals?

There is not a single or simple solution to this problem. The three patterns in this paper take a high-level approach to an answer. Together, they create a framework in the solution space spanned by people, process, and technology that provides a structure for adding more detailed patterns (see Appendix A) to build an organization tailored to distributed development.

People – The Global Society
Growing a unified workforce from a large number of individuals with a diverse background who are distributed on several geographical locations.

Process – Ceremonies and Traditions
Developing a methodology and practices that improve the work situation for such a distributed workforce.

Technology – Equipment
Tools and technology that support working across multiple locations.

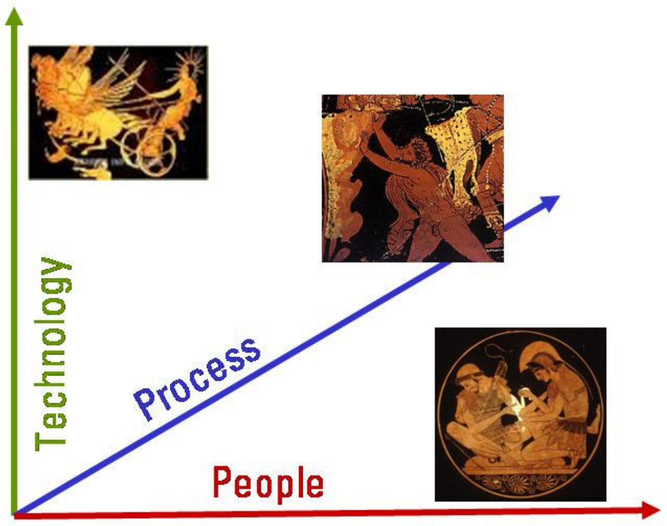


Figure 1: The framework for distributed development patterns

We are aware that the reader may ponder over how to apply these high level patterns in his or her own organization. The detailed patterns that together implemented *The Global Society* with its *Ceremonies and Traditions* and the *Equipment* chosen and used in our organization may need modifications and adjustments to be useful to other companies. We still hope that our experience and stories can be of help to others, and urge the reader to look for the reasons and the underlying values rather than the mechanical implementation of the patterns.

Acknowledgements

This paper was originally submitted for EuroPLoP 2006 and was shepherded for 2 iterations before it was decided by shepherd Joe Bergin and author together that it would need more work before being ready for a workshop. Joe's deep and thoughtful feedback is highly appreciated, and is still challenging the author during the continued work.

Warm thanks are also going to Didi Schütz who shepherded the paper for PLoP 2006. Without his good comments and questions, and not least his persistence in nailing down the author for discussions during EuroPLoP, this paper would not have been ready for PLoP this year.

The Global Society

For a variety of reasons, many project teams are today faced with developing software products while their team members are located geographically far apart.

To succeed with distributed product development you need a skilled and unified workforce worldwide.

***

Your organization develops software products with teams located across several geographical locations. Distributed development is expected as a norm for the organization. Due to various restrictions it is not an option to co-locate all team members at the start of a project.

When trying to develop software with distributed teams you face a number of tough challenges like communication problems, organizational differences, trust-respect-understanding issues, and other problems relating to an international workforce. **How can you develop an organization with the special characteristics and capabilities that enable distributed development?**

A common way for an organization to define development is through process, procedures, and tools. Although a good methodology and suitable tools are very helpful, the fundament for success is the value system and core understanding of distributed development of the people involved, because it is the values instilled in individuals that guide their actions.

Good and frequent communication between the team members on a project is believed to be essential for team performance. But the quality of communication for the distributed team will be affected by anything from language barriers, lack of overlapping work time, and a lack of good communication tools, to more fundamental organizational communication problems stemming from differences in organizational culture, envy between engineering locations, differences in business terminology etc. It is common in organizations with multiple locations to struggle with a fundamental "them" and "us" problem between locations that can be so severe that it cripples the productivity of collaborative efforts between the locations.

It is a relatively minor achievement to pull off a limited distributed development project as a one-time experiment with enthusiastic developers trying a new thing. But one successful experiment may mislead management to believe they will succeed with other projects. A sustained ability to successfully running large distributed development projects require enough respect for the challenge to take on the necessary investment.

Therefore:

Build a global society with a strong common identity. The key characteristics you need to ensure in your international workforce are:

- a strong feeling of identity towards the global organization rather than to any specific location, meaning that the society member truly feels as a recognized and valued part of the global workforce
- any team member who visit or transfers to another location quickly feels comfortable and is able to work at full speed because work environment and colleagues are already known
- a balance in culture and individual benefits that make any location attractive to the employee, and where no location stands out as being preferred or has a higher status

Several patterns work together to help you build this society. Note that each of these patterns cannot by itself generate the desired result.

> *Relocation and Rotation*
> A substantial part of the workforce is at all times located at another engineering center than their "home base". The transfer options are normally given to good performers and key technical personnel, and there is a certain natural rotation over time meaning that as many as 30-40% of the developers and managers have at some time worked at one ore more of the other engineering centers.
>
> *Balance of Nationalities and Minorities*
> The different nationalities should represent the amount of involvement in a country or region. The distribution of nationalities must be reflected in the management organization all the way up to the CEO.
>
> *International CoPs*
> Technical communities of practice are grown on the international level, networking through everything from blogs and wikis to workshops and conferences where members can physically meet.
>
> *One Class of Citizenship*
> Within what is feasible given local laws and regulations, each location comply with the overall company policies and give all employees the same opportunities and the same benefits. This applies to salaries and bonuses, personal development opportunities, international transfer possibilities, various leave of absence, retirement programs and pension programs, etc.

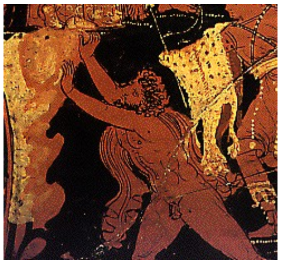
For any new location added to the organization, these patterns must be applied within a reasonable time frame. Within one year the new location and its people should be well underway to be fully accepted members of the society.

— oOo —

Over time, the workforce will develop a strong organizational identity. Through transfers and short-time travel to the other locations, strong networks will develop between individuals in the organization. Transfers are made easier by the locations being similar in how they manage their people. In the context of work, the primary identity for each individual in the organization is strongly tied to the society, and not to the local development group. *Ceremonies and Traditions* used by the society during work, team-building and social activities will further strengthen the group identity.

A major issue is the toll on employees and their families caused not only by extensive traveling and unusual work hours needed to communicate with other team members, but even more the strain on some families by international transfers. Another problem area is that the philosophy above does not encourage mid-career hires, which means that we lose out on very valuable knowledge and a broader perspective. We have also seen some resentment at a few locations caused by having an international workforce (taking away local work opportunities by filling positions with foreigners).

You are looking at organizational change so do not expect miracles to happen in a few weeks. Since it is not possible to apply some initiatives completely the same way in all countries, the organization must strive to find a way where the results from the employee perspective are perceived to be of similar value. Do not enforce the same implementation when the results may be far less beneficial to the employees for instance because of local taxes. The solution is specific, but very high level. There are a number of implementation details that need to be worked out. More important, the authority to implement the initiatives lies very high up in the company, and convincing the right people to implement this solution may not be possible in every company.



Ceremonies and Traditions

The software profession has developed work methodologies that have improved the success rate of software development and given the teams a clear idea of what to do and how to do it through every phase of the project. Most of these methodologies focus on team co-location.

Project teams doing distributed development need good work methodologies adapted to their situation.

***

Your organization develops software with teams located across locations. A *Global Society* is gradually growing through initiatives to build a unified workforce. There is confusion as how to best run distributed development projects. Some teams make it work, some not, and the reason why is not generally understood in the organization. New people are uncertain about what is expected.

What are the methods and techniques that work for distributed development in your organization? How can you make them generally known to the development teams?

There are a number of software development methodologies available. None of them focus on or are created specifically for distributed teams. Agile methodologies have become the preferred way of working for most developers, but have techniques that are difficult to do with remote team members.

Project teams face a number of new challenges; remote team members, different time zones, travel costs, communication difficulties, etc. and some teams seem to have overcome these challenges. For other teams in the same organization, possibly with less experience in distributed development, the path ahead is unclear and scary, and they have a hard time trying to understand what to prioritize and how to benefit from their colleagues knowledge.

There is limited availability of resources (internal or external documentation, consulting) for distributed development, and you need practices that work within your own organization anyway. Teams are reluctant to bother other development teams with tough deadlines to get help on their project.

Therefore:

Build ceremonies and traditions for distributed development by identifying core activities, workflows and practices that your successful projects apply. Disperse these through the development organization. It is very important to include team-building, celebrations and award events, and social activities since these have a big impact on your team dynamics. You will typically need to go through the following phases:

1. Mining for knowledge
   Interview project teams doing distributed development. Run brainstorming/workshops. Study literature and if possible study other companies doing distributed development. Read retrospectives reports. Look for people oriented practices as well as for project practices.
2. Documenting the knowledge
   As your understanding matures, you will want to document the key practices and formalize training. You can document in various ways (patterns, best practices database, training materials, stories, etc.).
3. Spreading the knowledge
   You can do this through training classes, lunch & learn, workshops, written materials, publishing retrospectives results, bulletin boards, etc. Experienced project managers can coach other projects (no extra overhead and all coaches have hands-on experience with the methodology).

Needless to say, the above phases need to be iterated as the organization is improving and growing new practices. These activities must cover and involve all members of your *Global Society*.

When going through this activity in our own organization we identified a number of patterns that formed our initial set of patterns for distributed development and that can be found in the appendix. These include *Early Bonding*, *Together*, and *Short Engagements* for face-to face interactions building trust and understanding, *Team Space* to accommodate visiting team members, *Iteration Connect* and *Completion United* for iterative development with user focus, etc.

The practices form a whole, but this must be promoted in a way giving freedom to the teams in selection of the practices. Work on promoting the practices that seem to work best for many teams, but do not establish a process police and a detailed process that all must follow. Focus on the benefits by using a specific practice (i.e. focus on the value system and not the motions).

As additional advice: from our observations, it looks like distributed projects that apply agile development methodologies are overall more successful. This may seem strange given the strong CMM push in India, and the focus in adaptive methodologies on small teams and strong, frequent communication (and preferably co-location in team rooms). Understanding the values of team interaction in agile methodologies, the teams have shown themselves creative in trying to emulate the same in the distributed setting. In this context, the technology and tools that make up your *Equipment* can be helpful if applied in the right way.

— oOo —

Eventually the organization will have established a solid set of practices for the *Global Society* that guides them during the software development. Most likely, this activity will also identify the tools and technology that are needed for the practices. There will be local variations, but essentially a member changing project or changing location will know the basic *Ceremonies and Traditions*, and the *Equipment* supporting them, and feel at home. You will likely find that several of the practices (especially if you are smart and include people-focused activities) enforce the team identity and contribute to strengthen the *Global Society*. Essential patterns for us are *Social Funds*, *Together* and *Early Bonding*.

You are establishing and growing practices that initiated in the development community, which is essentially good. But there is a potential risk that they conflict with “official” development process in the organization and thereby become impossible to implement on a broader basis. Another problem is to know if a certain practice really is beneficial, and if it will work for other teams and in slightly varying contexts.

New members of the society must be trained as they join. Never declare victory and think that the set of practices should stay as is. It must keep evolving as the world evolves, as new society members join with new ideas, as you can benefit from development in the software industry at large. And keep flexibility in work methodology for the individual teams, there is no “one size fits all” in development methodology.

Equipment

The software profession has developed work methodologies that have improved the success rate of software development and given the teams a clear idea of what to do and how to do it through every phase of the project. Most of these methodologies focus on team co-location.

Project teams doing distributed development need good work methodologies adapted to their situation.

***

Your organization develops software with teams located across locations. A *Global Society* is growing from a number of initiatives, and *Ceremonies and Traditions* for distributed development are getting established.

There is a lot of technology available in the market today that could be useful for distributed teams. How do you choose what is right for your organization?

Sophisticated tools are usually attractive. But some sophisticated tools are dangerous in that they remove the need for thinking or severely restrict your freedom of action. More advanced tools that support development workflows can be of great help, but there is always the danger that they force you to work following their logic and thereby breaking the practices you desire to apply.

Some of the technology is expensive to buy and/or use (like video conferencing in China). But communication between remote centers depend on technology, and going for cheap solutions for core needs may severely affect team productivity and the final product quality.

You can easily overdo the emphasis on equipment, thinking that the tools solve your underlying problems. Simple tools tend to give the most benefit for the investment.

Therefore:

Select the equipment that fulfills your needs in building the *Global Society* and enhancing the *Ceremonies and Traditions*. The essence is to provide the team with the necessary technology for communication and development, and to make sure they are all using the same versions and are following the same procedures in how the technology is used.

Our experience was that distributed development team will at least need the following set of technology with some key characteristics as defined for each item:

1. Appropriate communication tools
   International cell phones at least for key team members, access to video conferencing

equipment, and laptops or home computers and broadband at home (especially when frequent communication at odd hours) is a good investment for team productivity.

2. Collaboration web space (secure) and chat option
   All key project information must always be available to any team member, and a project team space on the web is a good solution.
3. Unified development environment
   Every location has the same development environment and versions installed (compilers, Visual Studio, modeling tools etc.) to ensure easy collaboration and code and documentation consistency.
4. Common code repository and baseline
   Working on different versions of the code base will mean tough integration issues and ample opportunity for incompatibilities and wasted development effort. So a global version control system for the code base is an absolute must. All team members should develop on the same baseline, with frequent check-ins and good unit testing to ensure there is always a working baseline. If possible, a system with incremental builds should be chosen.

— oOo —

The tools will not be why the project succeeds. The people will be why. But by providing the right technology to the team you can simplify their work and minimize the effect of not being co-located.

You may need to overcome local IT limitations to ensure upgrades and availability and bandwidth is the same and good for all developers. This is especially the case for developers working part time from home. You may also have cost issues providing the same level of technology to all team members.

Do not force technology on the teams, but let them chose from a list of available options that you can afford to provide for them. Every team will have different priorities. And never think that technology is a replacement for thinking! Identification of the right *Equipment* should be driven by the *Ceremonies and Traditions* that you want to establish and mature.

How the Patterns Evolved

As mentioned in the introduction, a distributed team is not a new invention. People have collaborated on research and product development projects internationally for decades if not more. Several years ago I was working for a Norwegian company as a project manager with projects both in Norway and in the UK, without considering that to be a problem. Traveling was easy and not expensive, so we would fly over on a regular basis, and communicate frequently by phone and e-mail.

Then a few years ago I got involved in projects running between the US and Asia. What had been smaller issues now escalated to become major obstacles. We were hiring a good number of young developers with no knowledge of our business. All the seniority and expertise was sitting on the US side. The cultural differences as well as the language itself were issues in communication. The time zones gave us no overlapping work time, which meant full day delays in exchanging e-mails, and caused very early/very late presence for video conferences and phone conversations. Travel was both time consuming and very expensive.

Luckily the people involved from both locations were enthusiastic and wanted to make it work. We were building up a new engineering center to have presence in the Asian market, so this was not a situation where people in the US felt that their jobs were threatened. Although stumbling from one problem to the next the teams found ways to collaborate and be productive. As some of the project managers moved on to other activities, we started to capture their experience to benefit future teams.

In the beginning, the project organization was informal and not so well planned for. Developers in Asia were assigned to teams in the US, but 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 that 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 heading 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.

Several modifications were done to improve this situation. It was clear that we could not continue to run projects with resources that were managed by two set of managers with different objectives. A side effect of this was also that funding was distributed on the locations resulting in a lack of accountability on the financial side. Several of our patterns reflect the established changes in the organization. These are all on a level that requires decisions from engineering center managers, i.e. they are outside what the project can control directly.

First of all you need *Commitment from All* in the development organization to ensure that the teams have the structure and support they need. A *Single Point Organization* gives clear decision making and responsibility on all levels in the organization independent on physical location. We now have a pure project organization, where each product delivery is defined as *One Project* with a project manager who has full control of the assigned resources.

The *Accounting Model* is set up to report spending versus funding on the global project, and each project is using a clearly defined *Communication Strategy* to inform its stakeholders about progress, achievements and upcoming issues.

At the start of a new project, the manager should if possible create a team of carefully *Selected* team members. She will need people who are flexible regarding work hours and traveling, and if she can find developers with previous experience from distributed teams that is of course a big benefit. A positive attitude to being part of a distributed team is another criterion. Even with some experienced people on board, the team needs training on distributed development to be well *Prepared*, and include training to build *Culture Awareness* in the team. Being together as a team for this training as well as for early planning is a good way to start the *Early Bonding* of the team. Make sure there are *Social Funds* available to the team from the beginning for social activities. If possible find an experienced *Mr. Mentor* to guide the team. *Short Engagements* before or early in the project should be considered to increase the team trust. Your team will benefit from a *Common Information Infrastructure* and a *Common Development Environment* from the start of the project. You may also want to establish the role of the *Mr. Connector* to handle project knowledge.

During the project the team should get *Together* with an *Iteration Connect* (we found this to be very powerful) at their various locations frequently (minimum every 2 months). To accommodate these periods we created specially designed *Team Space*. Between the physical meetings the team will have *Smart Meetings* maybe several times per week. Make sure you give people the *Flexibility* to balance this effort with their personal life. To establish a clear process for *Conflict Management* is most likely a good investment. Remember that *Risk is Everyone's Job* and risk needs to be evaluated and re-established at every iteration.

Another practice we have found to be incredibly efficient is to do a *Completion United* with our user community at the end of the project. Both during the project and in the completion phase it is very important to give *Full Credit* to all involved, not only the core team members.

Some other practices to consider is to create and keep going a *Team Terminology* not least for new team members joining later in the project, and to try out new ideas with *Pilot Solutions* before making large commitments and investments. Also remember that your development methodology is a *Living Process* so keep learning from your own teams and from the industry in general.

To establish the framework for an organization that can apply these patterns and sustain distributed development over time we suggest to work seriously on growing the *Global Society* with a strong common identity (*One Class of Citizenship),* created by establishing a *Relocation and Rotation* mechanism, with a *Balance of Nationalities and Minorities* and active and exciting *International COPs.*

As a final piece of advice we suggest being very clear on why your organization want to embark on distributed development. It really increases the risk for a project, and fundamentally should be avoided if possible. If it is necessary then set up a *Benefit Target* from the start to be clear on why the organization has chosen to work this way and what they expect to achieve.

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. “Can absence make a team grow stronger?” by A. Majchrzak, A Malhutra, J. Stamps and J. Lipnack, Harvard Business Review, May 2004

4. “Interaction Patterns of Agile Development” by Jens Coldewey, 2004

5. “Capable, Productive and Satisfied” by Paul Taylor, PLoP 1998.

6. “201 Principles of Software Development” by A.M. Davis, McGraw-Hill, 1995

7. “Culture Clash – Managing the Global High-Performance Team” by Thomas D. Zweifel, ISBN 1-59079-051-0, Swiss Consulting Group 2003

8. “Global Teams – How Top Multinationals Span Boundaries and Cultures with High-Speed Teamwork” by Michael J. Marquardt and Lisa Horvath, ISBN 0-89106-157-6, Davies-Black Publishing 2001

9. “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

10. “Trust within Global Virtual Teams” by Olivier Chavaren, ISBN 0-595-27577-X, iUniverse, Inc 2003

11. “Virtual Teams – Reaching across Space, Time, and Organizations with Technology” by Jessica Lipnak and Jeffrey Stamps, ISBN 0-471-16533-0, John Wiley & Sons 1997

12. “The Manager’s Pocket Guide to Virtual Teams” by Richard Bellingham, ISBN 0-87425-615-1, HRD Press, Inc 2001

13. “Global Software Development – Managing Virtual teams and Environments” by Dale Walter Karolak, ISBN 0-8186-8701-0, IEEE Computer Society 1998

14. “Managing Virtual Teams – Practical Techniques for High-Technology Project Managers” by Martha Haywood, ISBN 0-89006-913-1, Artech House 1998

15. “Working Virtually – Managing People for Successful Virtual Teams and Organizations” by Trina Hoefling, ISBN 1-57922-032-0, Stylus Publishing 2001

16. 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

Appendix: Pattern Thumbnails

The short descriptions below include Patterns (P) based on our experience with distributed teams, as well as Proto-patterns (Pp) based on recommendations made to better manage those distributed teams – and currently being tried throughout our organization.

1. **Relocation and Rotation (P)**
   You need an international workforce with a strong common identity that collaborate well and that is not having a “them” and “us” mentality.
   Relocate a part of your workforce on a rotational basis to other engineering centers. This will build relationships and trust across locations, and enforce common work methods.

2. **Balance of Nationalities and Minorities (P)**
   You want a workforce that is representative of your business involvement worldwide, and where individuals are equally respected independently of their background.
   Hire with a clear goal that the different nationalities represent the amount of involvement in a country or region. The distribution of nationalities must be reflected in the management organization all the way up to the CEO.

3. **One Class of Citizenship (P)**
   You want a workforce with a strong common identity, where individuals are equally respected and feel equally valuable to the organization.
   Make each location comply with the overall company policies and give all employees the same opportunities and the same benefits within what is possible given local laws and regulations.

4. **International COPs (P)**
   You need to fully benefit from the knowledge and creativity of your whole workforce to stay competitive. You want the participation of every individual, and that they actively participate and collaborate with other experts within their application domain.
   Grow and support technical communities of practice on the global level. Provide means of networking through blogs and wikis, and enable workshops and conferences where members can physically meet.

5. **Benefit Target (P)**
   The success of distributed teams is heavily discussed. Results are often questionable, and the drawbacks often by far outweigh the achieved benefits.
   To be able to keep the focus on the expected benefits and to determine achieved improvements, define 3 major targets for the expected benefits that are not in conflict with each other. The benefits must be specific to the organization and allow the organization to determine afterwards if each targeted benefit was reached.

6. **Commitment from All (P)**
   Even if the project is distributed over several locations, the management interest and attention may be stronger in some locations. This may lead to lower priority and lack of resources in the other locations.
   Any engineering center involved must have a clear commitment to the project. In the initial phase, a clearly defined involvement must be agreed between the involved centers.

7. **Single Point Organization (P)**
   Running projects in a distributed way often results in confusion and frustration for the involved parties. Conflicting objectives and unclear authority and responsibility distribution are just examples.
   The complexity can be reduced and controlled by a clear and communicated organization. This is more than defining the roles and responsibilities: it is key to ensure that the decision-making authority lies with only one manager at each level in the organization.

8. **One Project (P)**
   The daily life of team members will be influenced by the location. Developers may find conflicts between local objectives and what is expected as product deliveries.
   Create a clear project identity across the centers. All team members share the same team objectives.

9. **Communication Strategy (P)**
   Distributed teams may have a more complex set of management and stakeholders, and how/when/whom to keep informed may be a challenge.
   The team needs to work out a clear communication strategy that lists who to inform, what each are expecting, and the communication means.

10. **Accounting Model (P)**
    A project team depends on efficient accounting "services" in the organization to keep track of spending versus funding. These services are rarely set up to support a team on multiple locations.
    Set up a meta-level model of the accounting that defines how to manage the project financial status, and assign a clear owner within the accounting organization.

11. **Common Information Infrastructure (P)**
    A project team must share information across the complete team, not based on physical location.
    Set up an information structure for the global team that enable consolidated reporting of status and easy sharing of information.

12. **Common Development Environment (P)**
    Distributed teams have the same needs for baseline management as co-located teams. Keeping the code-base in sync requires immediate updates at all locations.
    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.

13. **Culture Awareness (P)**
    All cultures have their unwritten rules, and there are many examples of collaboration problems that stem from a lack of understanding of ethical and behavioral code. Note that the culture aspect includes organizational culture!
    Since we are not all born social anthropologists, some specific training may be necessary to learn to "read" team members from a different culture, and to ensure respect and good working relations.

14. **Selected (P)**
    Working on a distributed team is demanding, and requires flexibility in work hours, ability to travel and stay at other locations, and good social skills for the collaboration within a very heterogeneous team.
    Team members should be screened for personality and the team carefully built over time for maximum cohesion.

15. **Prepared (P)**
    Distributed teams have some added challenges, especially related to communication.
    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.

16. **Social Funds (P)**
    Team members cannot be expected to privately pay for social team events. These events are important for the bonding and thereby for team efficiency.
    Make sure funding is allocated to build personal relations between the team members (team building, celebration of achieved milestones etc.).

17. **Mr. Mentor (Pp)**
    There is much to learn from colleagues that have managed distributed teams before, but this learning may need some support from the organization. Each individual does not necessarily know who to ask, or feel that they can “bother” colleagues repeatedly for advice.
    By assigning a mentor with experience in distributed development to a new team, ongoing and on-the-job learning is made possible.

18. **Mr. Connector (Pp)**
    Distributed teams struggle with keeping on the same page between locations.
    Instead of everyone on the team trying to be on top of all ongoing issues, designate a team member at each site to manage the flow of information, and who knows how to get answers to different kinds of questions.

19. **Living Process (P)**
    A shared development process adapted to the team’s needs becomes even more important with the increased challenge and need for formality.
    Spell out the fundamental values, methodology and techniques used and keep it up-to-date for all team members (revisit at each iteration).

20. **Risk - Everyone’s Job (P)**
    Risk management is a very important area to cover well in a distributed team.
    The project risks must be globally understood and tracked, and all team members need to have the ownership of risk.

21. **Early Bonding (P)**
    Trust and personal connections need to be built early and are key to have a functional team.
    Emphasize on building the team from the start of the project, and include social events.

22. **Smart Meetings (P)**
    If co-located, you can do frequent progress meetings, and have spontaneous get-togethers when needed. In a distributed setting meetings need to be organized up front. But your communication needs are still there.
    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 (several times per week) to avoid locations drifting apart on issues.
23. **Together (P)**
    Even with Smart Meetings, communication suffer by not being co-located.
    Make sure the team physically meets frequent enough during development. For 1-2 year software projects, we have come to believe a 6-8 week frequency is the best compromise between all effects of traveling and the need to meet, each time staying together for about 10 days.
24. **Iteration Connect (P)**
    The most busy communication phase in a project is at the end of an iteration cycle, plus during the iteration assessment and the start of a new iteration.
    By synchronizing the length of the iteration (or a multiple of iterations) with the frequency of the physical team meetings we found the meeting time was spent most effectively.
25. **Completion United (P)**
    The final phase of a project is again a time where good and fast communication is important. System testers doing the final verification face significant problems when the team is distributed.
    We have found it amazingly effective to bring the development team together with all the testers and stakeholder/user representatives on a single site for completion.
26. **Flexibility (P)**
    The load of communication with other locations that frequently have to happen outside work hours is a big load on employees and on their families. The workday often gets very long, in reality working one normal day plus an extra shift early or late in the day.
    To balance the effort and get closer to a normal workload, redefine the team's work hours and give the team members flexibility outside common team time.
27. **Short Engagements (P)**
    Short-term assignments at the other location will enable team members to get to know other team members better.
28. **Conflict Management (Pp)**
    Choose and publish a pre-defined way of managing conflicts, possibly involving an independent party outside the core team.
29. **Full Credit (P)**
    It is amazing that this one needs to be written down, but experience tells us to do it:
    Make sure all team members get credited for success.

30. **Team Space (P)**
    Local team rooms need to accommodate visiting team members, VC equipment need to be easily available etc.

31. **Team Terminology (P)**
    Make sure you do not get local “dialects” and that everyone understand the lingo of the problem domain.

32. **Pilot Solutions (P)**
    Many of the practices we recommend have a certain cost associated, although we firmly believe they pay back multiple times over. But, there is management to convince.
    To gain experience, try out a new solution on one team first. Be sure to record the results well.