Managing Risks in Global Software Engineering

Authors

Christof Ebert
Vector Consulting Services
Ingersheimer Straße 24
70499 Stuttgart, GERMANY
christof.ebert@vector.com

Bvs Krishna Murthy
Alcatel-Lucent India
94/95 Thiruvika Ind. Estate, Guind,
600032 Chennai, INDIA
Bvs.Krishnamurthy@alcatel-lucent.com

Namo Narayan Jha
Alcatel-Lucent India
94/95 Thiruvika Ind. Estate, Guind,
600032 Chennai, INDIA
N.N.Jha@alcatel-lucent.com

Abstract

Globally distributed software development poses substantial risks to project and product management. Not all eventualities can however be buffered, because in the global economy, developing and implementing products must be fast, cost effective and adaptive to changing needs. Therefore, there is a need to utilize different techniques to effectively and efficiently mitigate risks. This article systematically introduces risk management in global software engineering (GSE) for product development, service and maintenance. Methods include using basic project, supplier and quality management techniques, process frameworks (e.g., CMMI), product life-cycle management, effective communication processes, SLA based escalation, competence management, and innovation management. A longitudinal empirical field study over several years from a captive SW center of a world-wide leading ICT company in India provides practical experiences and indicates how to effectively apply GSE risk management practices.

Copyright

1 Introduction

With global markets and a world-wide race for talent with software skills, global software engineering (GSE) model, is fast growing. GSE ranks as one of the key principles and drivers of software engineering – seen from a business, education and research perspective. It is the consequence of the rather friction-free economic principles across the entire software industry.

Companies continue increasing their global development efforts. Over half have significantly increased offshore work in the past years [1,2]. Fig. 1 shows for five different business sectors ranging from automotive and manufacturing to finance, consumer, ICT and health, how functions inside enterprises are already engaged into offshoring. Obviously IT functions have the highest degree of offshore capacity across the five sectors. However, the core of these sectors, namely R&D and engineering functions are at the steepest growth rate. It is not anymore that support functions and services are outsourced, as we were used to. Today’s emphasis is on globally utilizing research and engineering to develop products. GSE, at the crossing point of both the IT sector and the engineering function, naturally builds the spear head of this radical business change [3]. As a consequence, IT software and services exports are growing fast which we indicate for the two biggest source countries, namely India and China (Fig. 2) [4].

Many companies start global software engineering (GSE) due to perceived cost differences. An increasing share also takes advantage of skill availability in different parts of the world or quality awareness and process focus in some specialized industries and suppliers. Most of these companies engage globally active outsourcing companies to achieve fastest ramp-up of their globalization targets. After a while into that business they realize that savings are much smaller and problems are more difficult to cure than before.

![Fig. 1: Offshoring and globalization across industries](image)

What went wrong? Why do so many companies struggle to achieve the targets they initially set for their offshoring or GSE activities? What can we do better to make GSE a success? These questions have stimulated this article where we integrated experiences and empirical evidence from GSE over several years in different context and companies.

While the cost advantage and skill pool further fuels GSE, we should be conscious on the related risks that come on top to the regular project risks. Not knowing them and not mitigating them, means that soon your project belongs to the growing share of failed global en-
Managing Risks in Global Software Engineering

Concrete empirical results are provided for specific risk mitigation levers from an offshore ICT software center in India, where we have been looking to a total of close to hundred projects over the past decade. Our research method is an empirical field study with longitudinal data collection over several years [5]. The projects from which we extracted data are of different size between few person weeks and close to hundred person years. They include application software as well as embedded software. Being GSE projects by nature, the type of global engineering varies between a captive mode, where requirements and architecture decisions are done in the Western hemisphere, while development and test are done in India. There are also scenarios included where software is developed in a shared mode with distributed teams.

We have practically applied and evolved all mentioned guidelines both in own projects as well as working with clients in different industries. Due to the rather similar approach to GSE across industries with regard to selecting, managing and maintaining supplier agreement and processes, the mentioned results can be readily used in other companies.

Fig. 2: GSE export revenues from India and China

2 Global Software Engineering Risks

Increasingly companies realize that despite all the need and drive for GSE, it is a risky business. Not all software engineering tasks and projects benefit from GSE! Many turn it down after having tried. 20% of sourcing contracts are cancelled first year, 50% downstream for not reaching objectives [5,6]. Why do so many externalized development activities fail?

Savings from GSE is around 10-15 % of cost reduction after a 2-3 year learning curve. Initially outsourcing demands up to 20 % additional efforts [3,5]. How can such risks be reduced and thus GSE efficiency be improved?

As companies turn to GSE, they find the process of developing and launching new products becoming increasingly complex as they attempt to integrate skills, people and processes scattered in different places. While the classic centralized software development approach...
allowed solving problems in the coffee corner or around the white board, GSE teams are composed of individuals who are culturally, ethnically and functionally diverse. They work in different locations and time-zones and are not easily reachable for a chat on how to design an interface or how to resolve a bug that prevents test from progressing. Only 30% of all embedded software is developed in a global or distributed context, while the vast majority is co-located [1] due to the experienced risks with GSE. How can these risks are mitigated and thus flexibility be improved?

Risk management is the systematic application of management policies, procedures and practices to the tasks of identifying, analyzing, evaluating, treating and monitoring risk. On top of regular risk repositories and check lists, several specific risks must be stressed in global development projects. They relate to two major underlying risk drivers, namely insufficient processes and inadequate management.

To systematically identify risks and evaluate appropriate risk mitigation, we will look to the major drivers for GSE and then elaborate how they are impacted by specific GSE risks. Throughout our research over the past 10 years on global software engineering, we see four major drivers fueling the need for GSE, namely efficiency, presence, talent and flexibility. Fig. 3 provides an overview on these drivers and how they relate to specific GSE risks. Let us look to these four GSE drivers and related risks. GSE risk management will then be illustrated with concrete benchmarks in the following chapter.

1. Efficiency. Software and IT companies need to deliver fast and reliably while at the same time the competition is literally a mouse click away. Hardly any other business has so low entry barriers as IT and therefore stimulates an endless fight for efficiency along the dimensions of improved cost, quality and time to profit. GSE clearly helps in improving efficiency due to labor cost differences across the world, better quality with many well-trained and process-minded engineers especially in Asia and shorter time to profit with following the sun and developing and maintaining software in two to three shifts in different time zones. Directly related risks to the efficiency target are project delivery failures and insufficient quality.

2. Presence. Global R&D and software engineering has become part of companies’ growth strategies, because they are closer to their markets and they better understand how to cope with regional needs, be it software development or services. Such global growth is a self-sustaining force, as it demands increasing capacities in captive or outsourced software engineering centers. Directly related risks to the ambition of presence in distributed markets are instability with overly high change rates (requirement changes) and inadequate IPR management. Risk of requirement changes is specifically included here, as we observed higher rate of change in distributed teams when compared to co-located teams.

3. Talent. Computer science and engineering skills are scarce. Many countries do not have enough resources locally available to cope with the demand for IT and software products and services. Fueling this trend, many younger people got nervous with media-driven misperceptions about the danger of outsourcing and GSE for the entire field of software, that they decided to rather engage in fully different fields. The consequence is a global race for excellent software engineers. GSE is the instrument to provide such skills and handle the related supplier-processes. Directly related risks to the drive for global talent allocation are staff turnover rates, insufficient competences and wage and cost inflation.

4. Flexibility. Software organizations are driven by fast changing demands on skills and sheer numbers of engineers. With the development of a new and innovative product many people are needed with broad experiences, while when arriving in maintenance, these skill needs look different and manpower distributions are also changing. Such flexible demand can not anymore be handled inside the enterprise. GSE is the answer to provide skilled engineers just in time and thus allows building flexible eco systems combining suppliers, cus-
customers with engineering and service providers. Directly related risks to the flexibility goal are poor supplier services, lock-in, and distance & culture clashes.

Obviously not all companies that engage in GSE look to all four drivers with the same motivation. As a matter of fact, we even see a kind of trajectory where a vast majority of companies starts with efficiency needs (i.e., cost savings), and then moves on to presence in local markets, and only after these two forces are understood moves further to talent and flexibility. Also, it is clear that these four factors feed themselves. The more energy a company spends on for instance building a regional pool of skilled software engineers, the more it also considers how to best utilize these competencies to, for instance, build a regional market or develop new products for such markets.

In consequence not all companies will face mentioned risks in same depth and at the same time. In order to validate this risk list, we performed two types of analysis. First we did a profound analysis of GSE projects from ICT during the timeframe of 1996 to 2007. Looking to over hundred projects performed in ICT software centers in India, China, Brazil and Eastern Europe, we found a consistent pattern of the top ten risks. We then validated this initial list by looking to companies we are consulting with and also to published field studies [1,5,6,8,9,10,11,12,14]. Depending on the specific GSE-layout (e.g., with or without external supplier), the ranking list of these top-ten risks is as follows:

1. Project delivery failures
2. Insufficient quality
3. Distance and culture clashes
4. Staff turnover (mostly for captive centers)
5. Poor supplier services (only for outsourced GSE)
6. Instability with overly high change rate
7. Insufficient competences
8. Wage and cost inflation
9. Lock-in (only for outsourced GSE)
10. Inadequate IPR management

We will furtheron look to these risks in detail, sorted according to the four drivers as depicted in Fig. 3.

1. **Efficiency**: Speed to profit ahead of competitors.
   - Risks:
     - Project delivery failures
     - Insufficient quality

2. **Presence**: Global growth strategy. Learn from new markets.
   - Risks:
     - Instability with overly high change rate
     - Inadequate IPR management

3. **Talent**: Race for skilled people. Value creation happens where the skills are.
   - Risks:
     - Staff turnover
     - Insufficient competences
     - Wage and cost inflation

4. **Flexibility**: JIT organizational networks.
   - Risks:
     - Poor supplier services
     - Lock-in
     - Distance and culture clashes

Fig. 3: GSE drivers and related risks
A comparison of the above risk list with published research, specifically, Boehm [14], shows that except for items 3, 8 and 9 all other risk items are traditional project risks. However they combine differently and their effects influence each other. For instance, distance and culture clash results from developing projects in low cost countries with a corresponding reduction of experienced manpower in previous high cost countries. A conflict arises to manage restructuring on one side and transition knowledge from the same people to newcomers.

### 3 Risk Mitigation Patterns

#### 3.1 Efficiency

**Project delivery failures.** A standard risk for many projects, the risk of being late or over budget amplifies in probability and impact due to the intrinsic difficulties of managing a global development team.

As mitigation, project and team managers must be educated in estimation, planning, dependency management, uncertainty management, project monitoring and communication. The latter is crucial as experience shows that projects fail not because of unknowns but because of not willing to know or to communicate known facts.

We have seen from ramping-up internal software teams in Eastern Europe, India and China that solid processes not only accelerate introduction of GSE but also serve as a safety net to assure right training, good management practices, etc. We conducted a controlled experiment when ramping up offshore development teams in China. Our experience was that the building of such globally distributed development team was fastest and most reliable in the case where the demanding organization was on CMMI maturity level 3. The same was done with lower-maturity demanding organizations with the effect that the CMMI maturity level 2 organizations could manage with some external support, while the maturity 1 organization failed due to highly inefficient interface frictions and lots of rework.

We recommend maintaining an organization-wide risk repository with all project risks together with identified mitigation actions. At the start of a new project, the project manager has to take this organization wide risk repository and check what specific risks are applicable to his project together with any new items. The second, more a medium term approach, is to train all project managers. Using the CMMI and certifying in professional project management is an effective mitigation.

Another important and easy mitigation action is building on past project experiences. The key parameter for project success is schedule adherence. We suggest doing a periodic Root Cause Analysis (RCA) on completed projects and identify the key issues that contributed to delays. On these issues we can do a Pareto analysis to define focused actions for the most critical and repeating issues. Fig. 4 shows the impact of project delivery risk mitigation indicating a clear reduction in spread of schedule deviation, over years, as we increasingly apply our learning from previous projects in to next projects.
Insufficient quality. Working in different places or with teams in different organizations means, that many work products are moved across such places and teams with the risk of insufficient quality. Often the underlying rationale is that teams suppose that there will still be sufficient validation “downstream” so that quality deficiencies accumulate. Many global development projects suffer from a “ping pong” approach of work products being thrown back and forth due to poor quality. These stories repeat each other – independent of countries and culture. The designer in the Mexican team claims that the US-American specification was not good enough, while the integration tester in India kicks back the product because Mexico again delivered insufficient code quality.

The major risk mitigation to such repeated rework and increasing mistrust is to force quality gates on work product level. A work product is only accepted if it has the right quality level. Incoming work products are inspected at least in samples to check consistency and quality. Service level agreements and responsibility assignments reduce the “ping pong” effect because at the least it is clear who should do better.

We look to “Early Defect Removal” (EDR) as the measurement of defects found before start of test compared to total estimated defects and compare it to upfront-defined threshold [3]. It will provide warning signals so that corrective actions can be initiated, well before the product becomes due for delivery to customer. Having worked with this concept over many years in different companies, we observe a strong negative correlation of -0.9035 between mean EDR and mean schedule adherence for a set of around fifty projects in the timeframe of 2004-2007. Fig. 5 shows this trend over four year period which indicates that EDR is in-deed an advance indicator to reflect on quality and thus schedule variance of the product.

Fig. 5: Optimizing early defect removal during GSE activities contributes to improved schedule adherence
3.2 Presence

Instability with overly high change rate. Frequent changes create extra cost. Often being present in different markets with individual engineering teams means that each of the teams first of all looks to needs of the local market. When products and features are assembled, inconsistencies appear which cause late requirements changes. Global development amplifies such requirements engineering weaknesses that have in most companies long been present but could be camouflaged due to collocated teams. If specifications are insufficient, a remote team will either misunderstand or not accept them.

As a mitigation, GSE demands more reliable requirements and change requests. We recommend enforcing a rigid roadmapping process that provides sufficiently early insight into feature evolution and release planning. Teams will appreciate it with more anticipation and design for change. Global development demands more communication than co-located development. Specifications and documents must be carefully reviewed, because engineers on the other side trust to what is written. Establish for all distributed work packages a baseline and configuration management based on defined and proven quality entry criteria. The expectations and deliverables in terms of effort, deadline, duration and quality levels are to be clearly documented and agreed.

We observed that the number of changes to the requirements are highly (negatively) correlated to the content adherence, defined as percentage of features delivered at the end of the project to that of original required features at the start of the project. The correlation coefficient which we use as an early risk indicator and thus lead indicator for risk mitigation actions is -0.99 for a set of around fifty projects in the years of 2004-2007. A possible way to mitigate the risk of uncontrolled requirement changes is to closely monitor the requirement change index (number of changes to the features divided by total number of features required at the start of project, expressed as percentage). Fig. 6 shows how content adherence is related to requirement change index over years.

![ Requirement change index versus content adherence over years (Correlation = -0.99) ]

Fig. 6: Improved content adherence with GSE risk management

Inadequate IPR management. Intellectual property rights are a key success factor in software development. Mostly software is not patented and copyrights are not enforced equally in all regions of the world. Further risks are related to improper use of external software (e.g., OSS) and careless handling of confidential information (e.g., leaving contracts at printers, etc.).

As mitigation, make sure that the intellectual property is well-secured. Divide key assets into pieces and provide only fragments to each global team. Share according to strategic relevance. Reinforce copyright protection for external sources. A GPL-protected component included with your product might force you to fully disclose all the software of the product. This
can be handled both on policy and architectural levels. Most relevant however is that your teams get trained in copyrights and specifically on the traps related to open source software. Install and enforce effective policies for confidentiality, copyright protection and intellectual property handling and train all software engineers and managers on it. Rigorously punish wrong-doing and unprofessional behaviors in this critical domain.

At the same time do not underestimate the potential of new teams to explore the innovation. The first step of innovation is not knowing what is the right (read legacy) way of doing things and the distributed teams have an inherent, hidden potential to innovate. Creating awareness on how to identify IPR and potential patentable ideas, on-line forums to share ideas, voluntary moderators to guide raw ideas in to potential patents is some thing not to be ignored.

Fig. 7 shows how distributed teams have generated the patent proposals after awareness training and workshops are conducted in 2006 and 2007, respectively. The correlation between training and workshop timing and the number of idea generation can be seen as a quite strong, indicating the huge effect such awareness has on protecting IPR in GSE.

![Fig. 7: Impact of IPR training on patent generation and thus IPR protection](image)

### 3.3 Talent

Staff turnover. This is a specific risk especially in Asian countries due to abundant job opportunities in the respective economies. It is a generic risk whenever GSE has no clear integration with an organization’s overall engineering strategy and career paths, such as having a nearshore maintenance organization within a software company.

Regarding attrition we have to apply two parallel mitigation strategies. First, it is clear that attrition in certain places of the world is higher than, say in western Europe. So we have to cope with it and prepare to learn and live and deal with attrition in advance. This means advanced planning of buffers, long term retention mechanisms like loyalty bonus etc. Buffers could be foreseen if engineers’ unavailability exceeds certain thresholds. Note though that such buffers immediately impact the bottom line and should be carefully pooled to serve several projects.

Second we should measure attrition and its impact factors in order to control and limit staff turnover. We recommend conducting periodic employee engagement studies from which we can learn and improve the working environment, which shall limit attrition to manageable levels. Based on surveys we then look into specific incentives to keep people, even in times where stock options are not the preferred instrument. For instance, international career paths and excellent individual development skills reduce attrition.
Insufficient competencies. This is a risk in each development project; however it is amplified by the bigger dependencies given the globally distributed team combined with less visibility on resource planning and skills availability.

For mitigation we recommend assuring global competence management and resource planning (e.g., with a multiproject management tool such as Primavera or similar) and a skills management on the level of detailed technical skills necessary for the projects. Note that competence management is not the same as above mentioned attrition management. It is however a necessary condition to reduce attrition.

We recommend managing competency needs in parallel to technical and project roadmapping. It is a strategic task in the hands of engineering management, not HR. Organizations spend huge effort in training employees but often does not correlate whether the training and improved skills reflect the business needs. In our case study we have converted the overall team’s competency and skills (C&S) range in to a single measurable number by linking available C&S and normalizing with required business needs. This single index is monitored for improvement. A direct consequence of skill management is enhanced retention of employees. We observed a strong negative correlation between skill index computed during one year and that of team’s lead attrition, i.e., attrition of the teams in next year. Fig. 8 shows the strong influence of team skill index measured at end of 2006 with that of full year attrition in 2007. The correlation factor within a community of close to 1000 engineers is -0.85.

Fig. 8: Correlation between skill index and attrition in subsequent year.

Wage and cost inflation. With the global fight for software engineering talent, wage inflation is a major GSE risk. For instance, salaries in India have increased by 14% pa between 2004 and 2007. The annual increase in most Asian countries is around 10% [1,2].

The primary mitigation is to carefully consider which regions to utilize and to make a profound business case and cash flow planning taking into consideration expected wage increases in different regions of the world. Where external suppliers are involved, evaluate upfront their business models and past cost evolution. Determine upfront which supplier size fits best to your own company size and structure. A big supplier with a small enterprise won’t fit, because the SME will not have the chance to make corrections once the contract is settled. Evaluate offers during the tender of a supplier contract with dedicated estimation tools such as QSM SLIM to compare and judge feasibility [3].
3.4 Flexibility

Poor supplier services. One frequent risk with third party suppliers is not meeting the expectations in terms of quality and delivery schedule. SLAs won’t help because once this point is reached, even escalation will not help much, because it will take too long and the product or service quality is already hampered.

Therefore the primary mechanism for risk mitigation related to insufficient supplier services is to carefully evaluate one’s own processes and those of potential suppliers – before engaging into GSE. As a client you should always consider the golden rule of supplier management: You pay for what you get. Don’t get trapped into contracts that look “cheap” and later bring tons of extra cost due to weak processes and insufficient delivery quality. Preconditions of any successful supplier management are good processes on both sides, i.e., for the client and the supplier. Insufficient client processes cannot be externalized. They will not scale up from a single site to several sites. Often those low-maturity processes can be handled in localized development without many overheads due to co-located teams, but will fail with globalization. From our experiences we recommend having a CMMI (development and/or acquisition) maturity level 3 on both sides – and being applicable for all impacted engineering and sourcing processes.

When still in preparation mode, negotiate for a fixed project cost, where price is fixed and linked to deliverables with specific quality targets, often including penalty clauses. A fixed price project will make the supplier proactive for performance as payment is linked to quality deliveries and not time spent. Though generally fixed price is at higher cost compared to time and material at the contract negotiation stage, our lesson learnt is at the end of the project, it turns out to be generally 10 to 15% less costly than comparable time and material projects.

Supplier management includes clarification on non-disclosure and related agreements before starting negotiation. Establish clear acceptance and liability rules following contracting and legal schemes of your headquarter base. For maintenance projects they also include clear SLA in terms of response time, solution time, percentage of return failures etc. Set up clear and measurable service level agreements. Ensure that this SLA contains all that matters for you in the contract. Insist on periodic reporting according to the SLA. From the beginning define thresholds that establish when and how insufficient performance will be escalated. Measure supplier capability or demand such measurement based on industry standards, such as CMMI. Relate value you receive from suppliers to the risk and cost of the delivered services or components. Implement contract evaluation after each single project.

Consider sufficient time and budget (resources) for training the supplier on your processes. A very strong training tool is the Scrum process with short team meetings every day where recent results and next steps are briefly reviewed. Any uncertainty should be brought up in such reviews, which should take not more than 15-30 minutes and can be conducted even per telephone conference across sites. Build a supplier program management to handle the necessary review and decision processes. Agree with your supplier review and acceptance processes to assure the right quality level. Installing such processes after the contract signature will create the perception of policing the supplier. You can ask third parties in case of questions or needs for escalation.

Lock-in. With GSE supplier competition on a global market, external suppliers often start with rather low rates and once the projects are sufficiently large clients might be forced to lock-in with them due to progress of product development and knowledge transition. In the least we may have to face increasing cost inflation.

The primary risk mitigation is to have multiple partners and distribute critical knowledge on two sources. Each one shall know that we have a choice to make and that will make the ex-
ternal suppliers to remain competitive. To improve efficiency and reduce effects of lock-in, global teams must use the same tools, methods and processes. It is worth the extra money for tools licenses, although in a low cost country the additional load on engineering cost can be 10-20% for the necessary design tools. Our recommendation is to avoid supplier-specific and ad hoc tools as they won’t scale up and can bring substantial issues if backups cannot be restored or contents are corrupted. Process improvements and best practices gained over years of experience in one engineering team need to be replicated quickly, in other engineering teams. Common processes and tools across engineering teams will benefit quick spread of lessons learnt / defect prevention actions across teams.

Lock-in goes beyond suppliers. Do not forget about risks related to certain regions of the world, where you might currently be locked-in. We also recommend maintaining flexibility in where you work and with which supplier. Instabilities can be caused by political turmoil as well as earthquakes, civil war or terrorist attacks. Don’t put all your global development into one single site. Consider distributed hosting of infrastructure and backups. Periodically test the restore mechanisms to a different new site.

**Distance and culture clashes.** Globally distributed software development is highly impacted by work organization and effective work split. Working in a globally distributed project means overheads for planning and managing people. It means language and cultural barriers [13]. It creates jealousy between the more expensive engineers being afraid of losing their jobs, while forced to train their much cheaper counterparts. The barriers to such harmonization and cooperation are not to be underestimated. They range from language barriers to time zone barriers to incompatible technology infrastructures to heterogeneous product line cultures and not-invented-here syndromes. An obvious barrier is the individual profit and loss responsibility that in tough times means primarily focusing on current quarter results and not investing in future infrastructures. Incumbents perceive providing visibility a risk, because they become accountable and more subject to internal competition.

As risk mitigation we recommend collaboration and communication. Collaborate across disciplines, cultures, time, distance, organizations. Communication starts before the GSE project is kicked off. Fears, hopes, barriers must be articulated. Assess your organization carefully on such distance and culture risks. This demands a fully new skill set, currently not taught at universities (e.g., managerial, teaming, sharing without loosing) [3]. Cultural sensitization, periodic workshop between clients and suppliers and networking between various teams has been the effective risk mitigation strategy. Provide space for engineers to share engineers’ emotions with team leaders openly. Establish early warning systems to “smell” upcoming barriers and fears.

Collaboration also means effective and efficient tools support. The exchange of information between sites must be carefully planned. The closer tasks and software components are linked, the more need for good data communication. Tasks with high overlap should not been done with too much time distance. Especially with a high work time overlap, online collaboration has high demands on fast, reliable quality of service for video, engineering tools and online collaboration. A change management tool is not enough because engineering demands collaboration on content and knowledge. Plain supplier management platforms as they are offered today for handling online market places and tenders are also insufficient due to their limitations in sharing engineering information. You will need rules and workflow support for documentation, design reviews, change management boards, etc. We recommend installing workflow management and online accessible project, work product and process information to assure proper knowledge management.
Conclusions

Over the past years GSE – both captive and with external suppliers – has grown at a rapid pace. GSE today is part of the software engineering business and discipline as is testing or project management. It is not anymore nice to have for cost reduction, but a need to have for sustainable growth and competitiveness. Independent of their size, software organizations have recognized GSE as a key driver to become more efficient, achieve presence in worldwide markets, have access to talent according to actual needs and still have the flexibility necessary to cope with changing demands and rapid technology changes. These four drivers determine the offshore strategy of software and IT companies.

GSE amplifies typical software project and product related risks, such as project delivery failures and insufficient quality. Worse yet, it creates new risks, such as inadequate IPR management or lock-in situations with suppliers. These risks must be identified in due time and have to be considered together with the GSE strategy and its operational implementation.

This article highlights the top-ten GSE related risks as we have identified them over the past decade in a multitude of GSE projects and situations covering four continents. They are not specific to an industry or company size, but rather to the underlying life-cycle processes and management practices. Based on Alcatel-Lucent’s experiences and enhanced with consulting work in several other globally acting companies, we show impacts of these risks and also how they can be effectively mitigated. For this paper we have studied the project results over years and analyzed the correlation between risk mitigation actions and project deliveries. This paper is a result of empirical study of historical project data together. Whereas 7 out of 10 experienced risks in this paper are common to already published research, 3 new risks are presented. We have verified the validity of risk mitigation actions by establishing a strong correlation with expected end results of projects. Our experience clearly demonstrated that applying the discussed risk mitigation methods, has resulted in improved delivery schedules as outlined in Fig 4 and 5.

As a conclusion and for practical usage the risks and our recommended mitigation actions are summarized:

<table>
<thead>
<tr>
<th>Risk</th>
<th>Mitigation Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project delivery failures</td>
<td>Professionally train all project managers. Apply best practices from the CMMI (DEV + ACQ) frameworks and target maturity level 3. Maintain an organization risk repository. Use lessons learned and root cause analysis reports from previous projects to avoid repetition of problems.</td>
</tr>
<tr>
<td>Insufficient quality</td>
<td>Implement and systematically follow quality gates at work product level. Establish and use quality indicators. Monitor and use early defect ratio as a warning sign of insufficient specification and code quality.</td>
</tr>
<tr>
<td>Distance &amp; cultural clashes</td>
<td>Establish open communication across multiple channels. Use workflow management and online tools. Have periodic workshops with teams. Apply online team-building if visits won’t work.</td>
</tr>
<tr>
<td>Staff turnover</td>
<td>Learn to deal with staff turnover by means of pooled buffers. Establish long-term retention models. Periodically conduct employee engagement surveys.</td>
</tr>
</tbody>
</table>
Poor supplier services | Establish a fixed price contract scheme with agreed supplier management and escalation processes. 
Evolve towards a partner model with the supplier. 
Train suppliers on required processes. 
During the ramp-up period, carefully train supplier management on escalation procedures and your own required quality level. 
Escalate carefully and step-wise and avoid the SLA hammer. 
Rigorously highlight insufficient quality, delays or lack of visibility.

Instability with overly high change rate | Review and sign-off of all requirements. 
Monitor and control the requirements change index.

Insufficient competencies | Establish and maintain a global competency and skill management process across the entire GSE organization.

Wage and cost inflation | Distribute work across regions and anticipate wage increases. 
Carefully protect against supplier lock-in. 
Evaluate your own and suppliers’ business models over future years.

Lock-in with supplier | Work with multiple partners. 
Distribute critical knowledge 
Establish common processes and tools. 
Maintain back-up and recovery mechanisms.

Inadequate IPR management | Systematically train engineering and management on IPR. 
Establish and rigorously apply a strong policy on IPR protection. 
Encourage innovation on all GSE sites and promote patents.

Validity of risks. Not all of the above risks are suggested mitigation actions may be applicable to all organizations. Wage and cost escalation will not be an issue for growing teams, as generally new recruits are at a less cost than existing average and per head cost will come down, even if wage cost is going up. Professional training like certification of project managers increases the risk of attrition due to better sellable skill level in market! Long term retention methods for attrition management will itself contribute to other risk of wage escalation. Similarly, the strong correlation observed between skill development and attrition might not be an universal phenomena, or there might be other overlying attributes impacting attrition more strongly. We recommend that organizations make an internal analysis to fine tune their approach.

As a general rule for risk identification in a specific GSE environment, we recommend setting up undesired scenarios, evaluate their probability to occur and decide for some 10-20 of those scenarios to take dedicated mitigation action. A majority is mitigated inside the global development project (e.g., common tools), while only a few must be part of the corporate risk strategy (e.g., handling supplier defection). Organizations should not worry about the number of 10-20 scenarios. They repeat in each of the organization’s respective GSE projects and will build a kind of checklist with dedicated and organization-specific mitigation strategies that are reused in each new project.

Our empirical research provides data from a longitudinal study across close to hundred telecommunication projects and products of different size and managed either in captive or distributed mode. We have showed with this data that specifically early indicators, such as requirements change rate, early defect removal and skill level, key risk related to project performance and attrition can be effectively mitigated.

Risk mitigation does not come for free. It depends on process maturity, technology complexity and managerial competences additional cost for mitigation must be foreseen. Each of the following risk mitigation activities accounts for a 5-10 % increase to project cost [5,6,11]:
- IT infrastructure, global tools licenses.
- Distributed project management and progress control.
- Coordination and interface management.
- Liability coverage, legal support.
- Training, knowledge management, communication.
- Supplier and contract management.

This being said, initial saving potentials will be substantially reduced if organizations are immature in their processes and in GSE management.

Fragmented tasks handled in several sites combined with inadequate engineering and management processes not only ruin the GSE business case but are to our experience the number-one-reason for cancelled GSE engagements across industries.

Organizations on CMMI maturity level 1 or 2 should not expect that global software engineering would yield much benefits. Instead it will reveal major deficiencies in processes and workflow, which create all type of difficulties, such as insufficient quality, delays, additional cost, cancelled offshoring contracts, demotivated workforce in both places (previous and new), and many more. The only viable alternative for such low-maturity organizations is to ramp-up the own processes before proceeding with GSE.

Looking to these risk patterns, there is two problems with GSE: Getting started and keeping going. However, with the needs, the rewards and the mitigation patterns that we have showed here, you will translate risks to chances and opportunities which is what they should be seen.

References

Company profile

Vector Consulting Services is the leading consulting company to improve technical product development. Numerous clients from automotive, aerospace, medical, ICT and transport embark on our professional solutions. Objective-driven improvements combined with pragmatic implementation are what we stand for. Functional safety, process improvement, efficiency increase – our know-how provides our customers with measurable competitive advantages. The consulting branch of the globally active Vector Group, we support our clients worldwide with powerful consulting solutions that cover the entire product life-cycle and its related processes and tools.

www.vector.com/consulting

Biographies

Christof Ebert is managing director at Vector Consulting Services. A trusted advisor for companies around the world, he supports clients to improve product development and product strategy and to manage organizational changes. Dr. Ebert sits on a number of advisory and industry bodies. Over the years he had set up several offshoring sites, performed due diligence assessments and supported numerous companies in improving their global software engineering and IT outsourcing programs. He serves on the executive board of the IEEE International Conference on Global Software Engineering (www.ICGSE.org) series, and teaches at the University of Stuttgart.

Contact him at christof.ebert@vector.com


Pre-order discount:
http://bit.ly/cSjZgD

Reader feedback on “Global Software and IT”

"Summarizes experiences and guidance, processes and approaches for successfully handling global software development and outsourcing. It offers tons of practical hints and concrete explanations on how to do it better."
—Michael Corbett, Chairman of the International Association of Outsourcing Professionals

"This book stands out as the best source of information on distributed software development. Seldom do we see a book with the concepts completely backed by the experiences and views of the experts in their fields. Software developers and managers benefit from the broad spectrum of topics and companies in the case studies."
—S M Balasubramaniyan, Vice President, Wipro Technologies

"This book is very well-written, up-to-date and packed with practical advice for every software engineer and manager working on global projects. I think it will become 'the' excellent source for the industry on the subject of global software development".
—Bikram Sengupta, Manager, Software and Services Engineering, IBM Research – India

"This book provides a superb introduction to global software and IT projects and products. Most useful are the many practical advice summaries that are nicely highlighted with concrete "how-to" in each chapter."
—Erran Carmel, American University, USA