

Optimising distributed software development using best practice and benchmarking techniques

An increasing number of software development heads are finding they are managing a highly distributed software development organisation, not as a result of an operational objective but as a consequence of commercial mergers and acquisitions, increased partnership activities, or enforced short-term cost reductions with a dramatic impact on the performance. Organisations generally experience a rapid increase in development complexity (Point 1 on Figure 1) following the gradual and extensive built-up of knowledge, processes, tools and resources (curve A). The highest level of complexity also correspond to the organisation's maximum maturity and optimum performance capability level (Point 2).

The transformation to a complex distributed development environment causes two significant changes:

- Decline of performance capability
- Erosion of development expertise

These two changes combined greatly impact the organisation's ability to identify and introduce the right changes to their development environment to reverse performance decline (Point 3) and address the rising complexity of the development environment (Point 4).

Before an organisation can embark on the search for the appropriate performance turnaround solution, they are advised to quantify their current performance capability and the complexity of their environment. This proves highly valuable when developing a business case for investment and measuring the return on investment following performance turnaround, as well as demonstrating to executives the underlying reasons for a decline in performance.

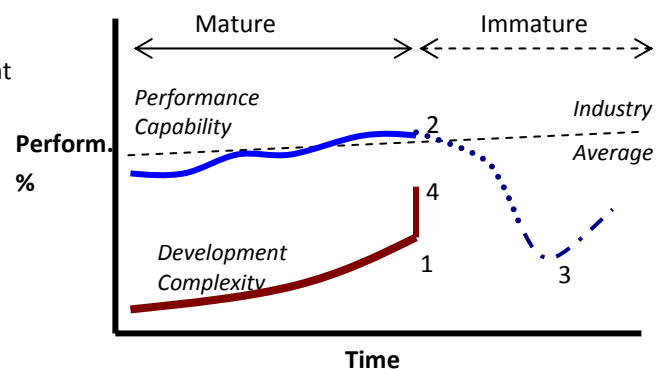


Figure 1: Enterprise Performance Chart

Performance capability

Performance capability is a measure of the percentage of projects that meet their defined objectives within a given period, e.g. month, quartile or project life cycle. The measurement used is the overall performance of projects which are delivered on time, to scope and on budget within the defined period.

Development complexity (processes, resources & tools)

A development environment quickly becomes more complex and performance declines for a number of reasons:

- Processes designed for a mature environment are manipulated to reach across multiple locations, with the subsequent introduction of temporary or ill-fitting processes.
- Resources become more varied, as new skills are introduced; however, these often have a limited understanding of the core philosophy, attitudes and beliefs of the organisation. Also, the overall technical and commercial knowledge changes, as do the ways the distributed teams communicate (possibly due to the use of a number of languages).
- Tools quickly become redundant, difficult to access or perform poorly across multiple locations. Incompatibility issues arise with data sharing and differences emerge amongst the tools being used.

Successful distributed software development

Successful distributed software development programmes recognise the importance and the need of an integrated and enterprise-wide approach to programme planning and execution. An increasing amount of subject matter evidence calls for performance evaluation and optimisation of such programmes to go beyond the activities of core software development functions, with teams collaborating increasingly with other enterprise departments and functions throughout the entire realisation process. Enterprises that quickly return to an optimum level of performance constantly achieve a high performance capability are those that exploit industry best practices in 10 core enterprise categories, such as innovation, communication, and knowledge management (see figure below). Overall, the enterprise’s capability is determined by a set of industry-wide subject matter expertise (Foundation) and predefined internal objectives (Tailored), which are blended to address the unique organisational characteristics and objectives of the change programme (i.e. reason for adopting a distributed software development model).

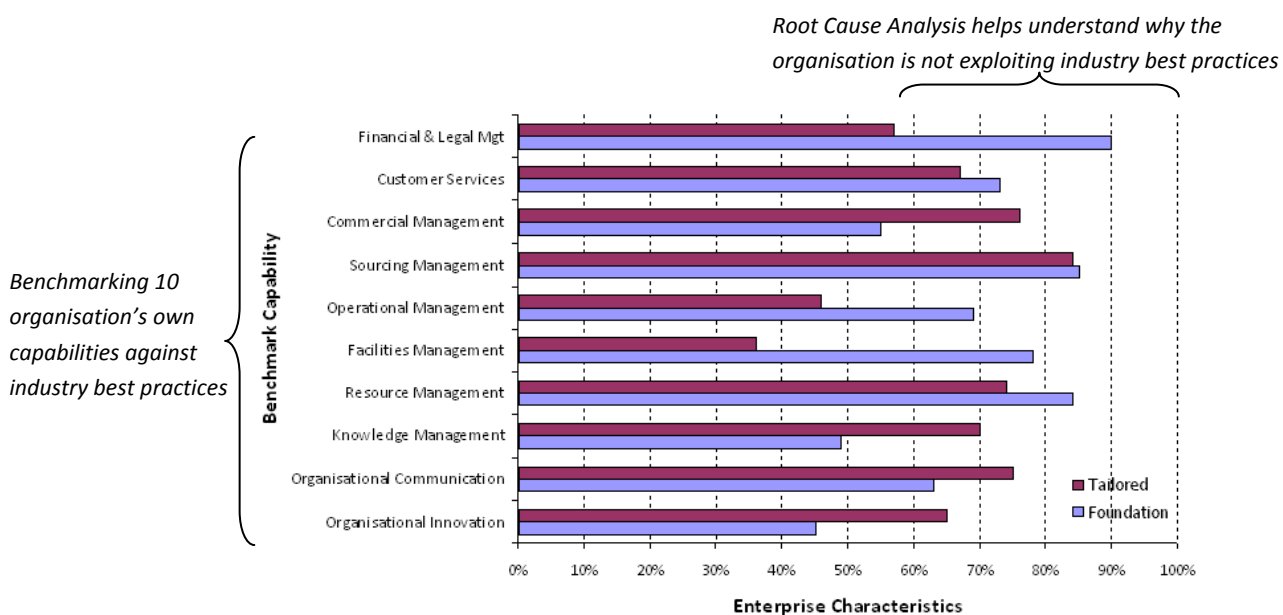


Figure 2: Enterprise Benchmark Performance Capability

Enterprise knowledge management life cycle

Returning the business performance capability to an historical high (Point 2 on Figure 1) requires a holistic approach to enterprise knowledge management, blending internal expertise with industry best practices from external sources to identify the most appropriate changes to the enterprise’s processes, tools and resources. Three distinct knowledge management techniques (i.e. Root Cause Analysis, Benchmarking and Best Practices Adoption) have an important role to play and it is therefore worth developing a greater understanding of them.

What is benchmarking?

Benchmarking is the process of comparing enterprise characteristics to other characteristics that are widely considered to be industry standard best practices. Essentially, benchmarking provides a snapshot of the performance of the business and understands where it is in relation to a particular standard. Although there are a number of procedural benchmarking

Figure 3: Electronic Benchmark Application

techniques widely used across the industry – including exploratory research techniques such as focus groups, in-depth research, quantitative research, surveys or questionnaires. The final choice is to establish how well the distributed organisation will be able to provide real-time, unambiguous, quantified and efficient input that can be collected and managed centrally.

All these points together indicate the use of an electronic survey or questionnaire as the best approach, as the analysis against predefined criteria (i.e. published best practice) allows the definition of a set of preconfigured responses that can quickly provide quantified responses. It is also worth noting that in a distributed environment, interview techniques and workshop forums are rejected because of the investment in time, management and travel costs associated to them. On the other hand, surveys or questionnaires that solicit and capture electronically observations or responses to a set of predefined questions have proven to be extremely productive.

What is root cause analysis?

Root cause analysis (RCA) is a problem solving method aimed at identifying the root causes of problems or events and correcting or eliminating them, as opposed to merely addressing the immediately obvious symptoms. In this situation, root cause analysis enables organisations to transform their understanding of how well they are exploiting industry best practices, e.g. 55% Innovation Management.

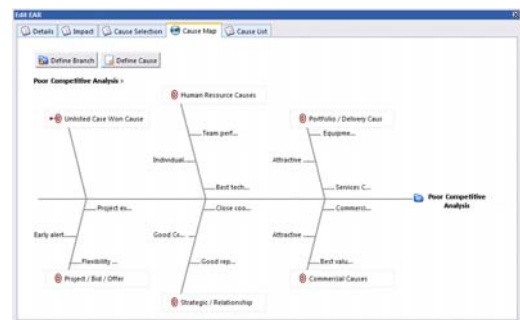


Figure 4: Electronic Fishbone Diagram

Similar to benchmarking, root cause analysis techniques are widely adopted. These include factor tree analysis, failure mode and effects analysis, fault tree analysis and fishbone diagrams (also called Ishikawa diagrams or cause-and-effect diagrams). In particular, preconfigured-electronic versions of the fishbone diagram (see figure 5) generally provide a highly efficient way of identifying the underlying weaknesses within organisations in the form of unplanned changes or poor benchmarking results. By providing a preconfigured fishbone diagram against each potential weak benchmark result enables online collaboration, which in-turn leads to real-time diagnosis of the underlying weaknesses rather than simply dealing with the symptom. This is also supported by the fact that this technique provides an opportunity to capture in real-time the associated impact and corrective costs of each weakness, and exploit it to prioritise best practice publication and use.

What is a best practice?

A best practice is a technique, method, process or activity that is believed to be more effective at delivering a particular outcome than any other technique, method or process when applied to a particular condition or circumstance. It is important to recognise that best practices emerge from a number of internal and external sources, which are constantly evolving from trial and error within different environments. As indicated in Figure 1, it takes a distributed organisation many years to tailor its processes, resources and tools before they can reach and hopefully maintain an optimum level of performance.

As the organisation reaches its performance capability optimum level, it will have developed a set of best practices, which manifest and are shared within an array of operational specifications, e.g. quality procedures, work instructions and specification templates etc. Unfortunately, consequently to their adoption by the distributed development organisation, the value of many of the existing best practices will have eroded. It is therefore important that new best practices materialise from an understanding of the value of existing best practices within the new environment and the need for them.

It is also vital to evaluate how the new best practices will be implemented, in terms of the new subject matter expertise needed to devise so. This important factor, which might have significant hidden costs, is typically overlooked when developing the original business case and planning the investment:

- Accessibility – best practices must be accessible by all employees, independent of their location
- Interpretation – whether introducing new best practices from a new external source or adoption existing best practices resources that will require tailoring
- Measurement – investment is needed to measure the value of individual best practices and prioritise their promotion and adoption

Capturing, analyzing and publishing best practices

An alternative and helpful way to evaluate at the knowledge management life cycle introduced earlier is to consider the best practice feedback loop (see Figure 6). The feedback loop enables the organisation to understand how individual best practices, either internally or externally sourced, have influenced performance and what further tailoring is needed to steer the organisation towards an optimum level of performance.

The feedback loop is designed to address 2 weaknesses commonly found within newly formed distributed software organisations. The first weakness is to treat each project as unique (i.e. as represented by the black hole), which means that lessons learnt (step 8) are not brought back into the life cycle to help improve the planning phase of new projects. The second and possibly the most important characteristics of any successful knowledge management life cycle is the need to constantly tailor best practices to the evolving landscape. Project goals and objectives change, resources are constantly evolving (step 3), as do the processes adopted by the organisation; it is therefore critical that the feedback loop is designed around the ability to measure and analyse all these factors to identify their influence on projects of different types and understand what impact best practices have to play.

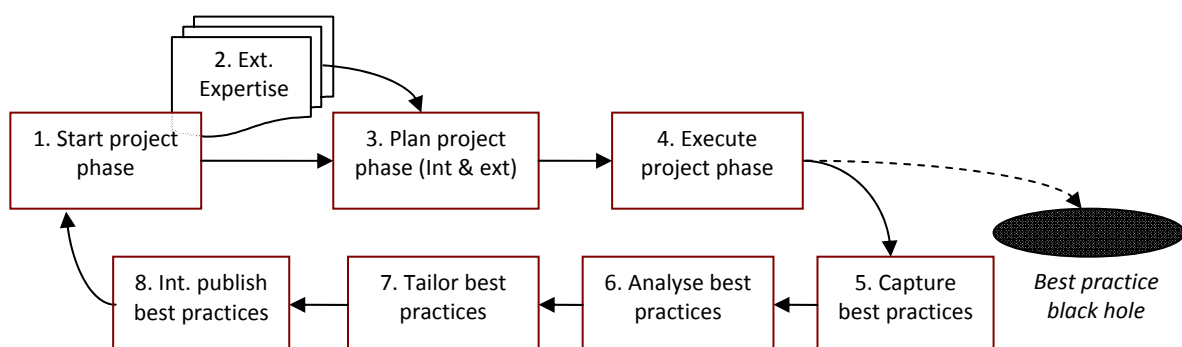


Figure 5: Optimising distributed development feedback loop

Whilst the introduction of external subject matter expertise (step 2) provides an important, short term focus for the newly formed distributed software organisation, this knowledge should be used to kick starts the performance turnaround process. As the appreciation for the best practices and the experience of the organisation increase, the understanding of broader enterprise factors and this analysis (step 6) helps further tailoring (step 7) from which individual internal best practices emerge.

Subject matter acquisition

As mentioned earlier, it is important for organisations to accurately measure their own enterprise capability and plan the most appropriate enterprise corrective measures. The corrective measures are designed to address underlying weaknesses in their existing best practices (using root causes analysis) and establishing what subject matter expertise they require is an important eEnterprise decision. Once this has been decided, they are then faced with the second critical decision: from where it is to be sourced?

As the figure below shows, there is a great variety of subject matter expertise sources available. A number of important factors should therefore be considered to make the right choice, as each source present different costs for the acquisition of knowledge, as well as transformational costs – i.e. tailoring to the organisation’s unique position, accessibility and published format.

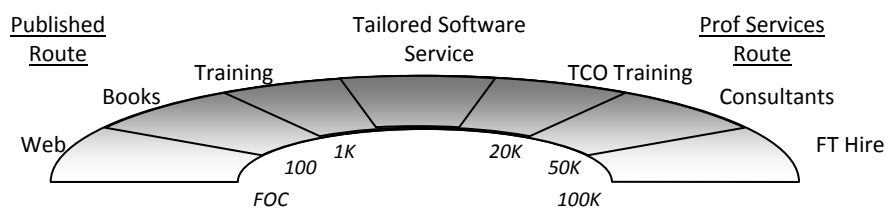


Figure 6: Spectrum of Subject Matter Expertise

A recent phenomenon is the emerging of online tailored software services that blend the right subject matter expertise with a host of online tools to aid the benchmarking process, enable root cause analysis and help identify and publish new best practices across a highly distributed software organisation. Tailored software services offer organisations, particularly distributed organisations, the following distinct advantages over the traditional published and service routes described above:

1. Knowledge management – All intellect is held electronically allowing direct, up to date and controlled access by everyone independent of their location
2. Change management - The ability to constantly tailor best practices based on newfound knowledge ensures everyone is granted access to the vary latest information
3. Interpretation - The effort required to interpret generic-published knowledge to address unique organisational and project characteristics is easy and allows complex relationships to be identified
4. Cost – T&L costs alone can lead to a very high consulting fee, if all locations are to engage in the life cycle
5. Economies of scale – Increase the depth and breath of the knowledge management process without significantly increasing costs
6. Performance measurement - Organisations can accurately measure current performance capability and closely monitor improvement

Performance optimisation to do list

Recommended performance turnaround actions for a distributed software development

1. Root Cause – Perform detailed root cause analysis for each project variation. Root cause analysis should be accompanied by a record of impact on project schedule (days), financial costs and resources (man-days).
2. Measure current organisational capability – Capture the necessary data to allow the creation of a chart similar to Figure 1. The table below will help you calculate the necessary values.
3. Source subject matter expertise – Establish the best source of subject matter expertise, convert into a format that enables accurate, undisruptive and efficient analysis.
4. Benchmark current organisational capability - Identifying where the organisation is currently not exploiting industry best practices by gathering input from all enterprise stakeholders.
5. Develop performance improvement plan – Work with published best practices and corrective measures to address most costly root causes.
6. Manage publish best practices guidelines – Determine most appropriate best practice guidelines.
7. Performance project tollgates – Ensure benchmarking becomes a routine part of project planning and execution.
8. Organisational changes – Start the implementation of the appropriate changes to enterprise resources, processes and tools.
9. Monitor performance – Make changes to current project quality procedures.

Calculating Organisational Complexity

Enterprise Characteristics	Parameters	Before	After
Resource - Number of resources (123) multiply by the number of locations development is performed	No of Resources = R	45	40
	No of Locations = L	1	2
	Resource complexity $RC = R \times L$	45	80
Process - Maturity of current processes I. Software development process for current landscape (i.e. plan-based - Agile) II. Enterprise-wide quality management system, (e.g. ISO9001)	Maturity, No of year SWD Loc_1 (15 - X yrs)	9	9
	Maturity, No of year SWD Loc_2 (15 - X yrs)		15
		9	24
	QMS maturity No of year = QMS Loc_1 (10 - X yrs)	5	5
QMS maturity No of year = QMS Loc_2 (10 - X yrs)	5	10	
	Process Maturity $PM = (SWD + QMS) * 2$	10	20
Tools - Amount of variation across the primary tools used by the software development environment. (0= no variation, 5 = all different)		1	3
	Complexity = (RC + PM) x TV	64	372