

EXPERIENCE BENCHMARKING @ DATAMAX

Benchmarking Workbook

Benchmarking accuracy depends on the quality of the data you provide. In this workbook you will find all the definitions you need to ensure that you are submitting comparable data. If you have any questions please contact us. Use this workbook to collect your software project data. Then fax this workbook to us at the number below. Keep this workbook for your records.

Project Identification Number: _____
(This will be attributed to you once you have confirmed your benchmarking report order.)

Start date: _____

End date: _____

The more questions you can answer about your software project, the more interesting and valuable your benchmarking report will be. If you cannot answer some questions, your report will include information about comparable projects, but no comparison to your project. For example, your answers to the 21 productivity factors are used to identify levers for improvement. If you do not provide data for these factors, we cannot determine what impact they had on your project's productivity. However, you will still learn about their impact on comparable projects' productivity.

There are 7 Obligatory Questions:

Total Effort
Total Duration
Total Software Size
Type of Development
Business Sector
Hardware Platform
Application Language

These questions are highlighted in red in the workbook.

EXPERIENCE BENCHMARKING @ DATAMAX

You cannot benchmark your project's software development productivity if you cannot answer the following questions about its effort and size. This is because software development productivity is defined as size divided by effort.

I. Effort

Effort is measured in hours. It is defined as the work carried out by the software supplier from the beginning of the requirement specification phase to customer delivery.

Total Effort in hours: _____

Total Duration in months: _____

	Effort (hours)	Start Date (d/m/y)	End Date (d/m/y)	Maximum Team Size
Requirements Specification				
Design				
Implementation (includes coding and unit testing)				
Systems Testing and Installation				

Maximum Team Size is the maximum number of individuals working during each phase, even if they were only working part-time.

How many hours does one person work in a month in your company?

One person-month = _____ hours

II. Software Size

Software size is measured in function points. The definition of size depends on the **type of development**.

- New Development: the software size is the function point count of the entire application.
- Enhancement: the software size is the function point count of the new features only.

We can accept Experience function points, Mark II function points, and IFPUG function points. If you submit IFPUG function points be sure to use the unadjusted FP count. Your benchmarking report will be based on the function point counting method you used.

Number of function points: _____ Function point counting method used: _____

If you have not counted function points don't worry, we can estimate the size of your software application using the KISS method developed by STTF. Your answers to the 28 Functional Requirements questions on the next page will be converted to KISS function points.

EXPERIENCE BENCHMARKING @ DATAMAX

28 Functional Requirements Breakdown Questions:

Answer all of these questions even if you know the function point count of your application. We use this data to compare the different kinds of functional features of your application. For an enhancement project, answer these questions about the new features only.

Six questions about navigation and queries, all interactive user functions without update facilities:

1. How many **starting icons** do you have in your application? _____
2. How many different **log-in and log-out windows** do you have? _____
3. How many different **menus** do you have in your application? _____
4. How many parameter **selection lists (drop-down lists)** do you have in your application? _____
5. How many **inquiry windows** do you have (showing contents of the database to the user on the screen, not updating anything)? _____
6. How many different **screens for starting report generation or calculative batch processing** do you have?

Three questions about interactive user input functions:

7. How many different **3-functional (create, update and delete) user input windows** do you have?

8. How many different **2-functional (create and/or update and/or delete) user input windows** do you have? _____
9. How many different **1-functional (create or update or delete) user input windows** do you have? _____

Four questions about passive user output functions:

10. How many different **formal datasheets** can you print out during an interactive session (no print screens!)?

11. How many different **paper reports** does your application provide? _____
12. How many different **text messages or e-mails** does your application write and send? _____
13. How many different **monitor screen outputs** do you have? _____

Seven questions about interfaces between this application and other applications:

14. How many **logical interfaces** does your application have with other applications? _____
15. How many different **messages to other applications** do you send? _____
16. How many different **messages from other applications** do you receive? _____
17. How many different **signals to a device** do you send? _____
18. How many different **signals from a device** do you receive? _____
19. How many different **batch records to another application** do you send? _____
20. How many different **batch records from other applications** do you receive? _____

EXPERIENCE BENCHMARKING @ DATAMAX

Three questions about data storage functions:

21. How many different **logical files** do you have? _____
22. How many different **entities or business classes (OO)** do you have? _____
23. How many **other logical record types** do you have? _____

Five questions about independent algorithmic functions required by the end-user:

24. How many different **independent calculation routines** does your application provide? _____
25. How many different **independent simulation routines** does your application provide? _____
26. How many different **independent formatting routines** does your application provide? _____
27. How many different **independent database cleaning routines** does your application provide? _____
28. How many **other** different **independent algorithmic routines** does your application provide? _____

EXPERIENCE BENCHMARKING @ DATAMAX

III. Main Classifiers

Several studies have found that software development productivity is strongly related to the following three factors: business sector, hardware platform, and application language. Your project will be benchmarked against projects that match on at least one of these factors.

Business Sector

The business sector of the customer.

- Banking, Wholesale and Retail Trade, Insurance, Manufacturing,
- Public administration, Telecommunication, Defence/Military,
- Services, Research and Development, Publishing, IT business
- Other, describe: _____

Hardware Platform

The hardware platform on which the application runs.

- Multi-platform, Mainframe, Mini Computer, Networked, PC
- Other, describe: _____

Language

Main application language used: _____

Other languages used (up to 3): _____

EXPERIENCE BENCHMARKING @ DATAMAX

IV. Situation Analysis

We need some background information about your project in order to explain why its productivity is higher or lower than comparable projects. Many project, process, product and people situations experienced during software development can affect productivity. These situations are described by 21 productivity factors in the Experience database. Each factor has five alternative values. The basic idea is that the better the situation, the more positive the rating. Check the situation that corresponds to your project.

General definitions:

- “--“ = Very bad situation, much worse than in average case
- “-“ = Bad situation, worse than in average case
- “+/-“ = Normal situation in the productivity point of view
- “+” = Good situation, better than in average case
- “++” = Excellent situation, much better than in average case

1. Involvement of the customer representatives

How actively the customer (can be interpreted as an end-user, purchaser or a defined market segment) participates in the development work.

- The customer is not interested or has no time to participate in requirements specification and the project in practice.
- Participation of the customer representatives is passive or rather formal. A small amount (less than 30 %) of the software functions are defined or approved by them.
- +/- The customer representatives participate in the project with satisfactory activity. Approximately half of the functions (30-70 %) are defined and approved by them.
- + The customer representatives participate actively. The majority of the functions (over 70 %) are defined and approved by them (all the most important functions).
- ++ The customer representatives are available when needed. Most of the requirements are specified and controlled by them.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

2. Performance and availability of the development environment

Performance level of the tool resources and equipment (tools, hardware, physical environment, network etc.) during the project.

- Shortcomings of the development environment are very common. Testing requires special arrangements because of operative production reserving all available resources.
- Shared equipment/machine resources. Delays in some work stages (for example compiling and testing).
- +/- Enough equipment/resources during the development life cycle. Reasonable workstations, software tools and physical sites for everybody.
- + Enough equipment and tool resources even for capacity peaks (efficiency, storage, response time etc... criteria fully met).
- ++ Dedicated, over-dimensioned development environments, in practice only for this project.
- Don't know.

3. Availability of IT staff

The availability of the software personnel during the project.

- Great problems in availability of professional software personnel for most tasks (lots of simultaneous customer and maintenance responsibilities, special know-how required).
- The team members are involved in some other simultaneous projects (also maintenance responsibilities). Low priority for this job.
- +/- The key members of the project team are involved in at most one other project in addition to this project. Possibility of overload and impact on this project.
- + The members of the project team are involved in this project almost full time. No special qualifications required. Some overload possible for key staff.
- ++ Qualified software personnel available when needed and they can participate fully on this project. Can handle high overload for short times during the project.
- Don't know.

4. Number of different stakeholders

Dependency based on total number of involved organisations and parallel dependent projects. Check the value in table below.

Number of organisational units involved	Number of parallel dependent projects				
	>5	4-5	2-3	1	0
more than 3	<input type="checkbox"/> --	<input type="checkbox"/> --	<input type="checkbox"/> -	<input type="checkbox"/> +/-	<input type="checkbox"/> +/-
2-3	<input type="checkbox"/> -	<input type="checkbox"/> +/-	<input type="checkbox"/> +/-	<input type="checkbox"/> +/-	<input type="checkbox"/> +
only 1	<input type="checkbox"/> +/-	<input type="checkbox"/> +/-	<input type="checkbox"/> +	<input type="checkbox"/> ++	<input type="checkbox"/> ++

- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

5. Pressure on schedule

The pressure on schedule, goal schedule compared to first estimate.

- Goal schedule either extremely tight or very loose. The target duration is either less than 50 % of first estimate or more than 100 % longer.
- Goal schedule clearly tight or clearly loose. The target duration is either 50-80 % of first estimate or 50-100 % longer.
- +/- Goal schedule slightly loose, the target duration is about 20-50 % longer than first estimate.
- + Goal schedule at realistic level, the target duration is less than 20 % longer than first estimate.
- ++ Goal schedule ambitious, moderately challenging, but not impossible. The target duration is about 80-90 % of first estimate.
- Don't know.

6. Impact of standards

The quality of the existing standards and procedures applied in the project.

- The standards and basic practices must be followed strictly, but they are unstable and will be changing during the project.
- Standards are partially OK, more procedures must be developed for some tasks. Not familiar in the project or the organisation beforehand.
- +/- Well-known but general standards that have been applied before. Some tailoring needed for most major tasks.
- + Detailed standards, which have been applied in the same environment for some time.
- ++ Stable and detailed standards, which are already familiar for the team. Flexibility in application and good control of use.
- Don't know.

7. Impact of methods

The use and quality of the methods to be exploited during the project.

- The project does not use any software engineering or management methods. (Traditional meetings, individual work.)
- The use of methods is beginning, traditional concepts (structural analysis and design, top-down design...)
- +/- Generally known methods used (structural analysis and design, conceptual analysis, ER-modelling etc.)
- + Methods are integrated at detailed level and they cover most activities and team work, support exists.
- ++ Methods cover the whole lifecycle and are tailored to satisfy the specific needs of the project, well organised support.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

8. Impact of tools

The use and quality of all kind of tools available for the project (programming, testing, documenting, project management etc.).

- Minimal set of basic tools, like editors, compilers and simple debugging tools.
- Basic tools, like interpreters, editors, compilers, debuggers, data bases, libraries.
- +/- Development environment, data base management system, support for most phases.
- + Modern tools, like CASE, project planning, application generators, testing tools. The standardised interfaces between the phases and/or tools.
- ++ Integrated CASE-environment, covers the whole lifecycle. All tools can support each other flexibly.
- Don't know.

9. Level of change management

The stability and predictability of the functional requirements in the project, maturity of the change management process.

- Continuously new requirements. No real contract for the project. More than 30 % of the functions are (expected to be) new or modified compared to original requirements.
- Some of the accepted changes are essential and have a clear impact on the application architecture. Return to previous phases and modifications to previous deliveries. 15-30 % of the functions are new or modified.
- +/- Changes to specifications occur, but they can be managed and their impact is minor (less than 15 % new or modified functions).
- + Some changes to specifications, some new or adapted functions, some minor changes in data contents.
- ++ No new features during the project. For example a pure conversion project.
- Don't know.

10. Maturity of software development process

Stability and conformance of the software development processes and lifecycle related activities in the project.

- Volatile requirements cause return to previous phases many times during the project. No means to eliminate that behaviour. No conformance with quality models.
- Return to previous phase may happen many times in different parts of the software. Basic practices conform mostly with internal procedures.
- +/- Some rework caused by continuous changes and late discovered defects. Some integration problems are expected during testing and release processes. Conformance with internal audits.
- + Stable progress, no return to previous steps, reviews find most of the defects early. Conformance with quality models.
- ++ Complete straightforward progress in all parts of the software, no defect or integrity problems expected. Strong tool support for all processes. Strong conformance with quality models.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

11. Functionality requirements of software

Compatibility with end-user needs, complexity of the requirements, level of integration.

- Virginal and complex application area, security critical, big (thousands of fp's) multi-tier system for various, multi-cultural users.
- New, interoperable application area with some complex features, demanding special understanding from the users and developers.
- +/- Partially automated, integrated application area, average size (600-1000 fp's) application with normal security requirements.
- + A largely automated application area, application has less than 5 interfaces with other systems, no specific security requirements.
- ++ Very mature, straightforward and easy application area, a small (less than 200 fp's) stand-alone application for a small group of users.
- Don't know.

12. Reliability requirements of software

Maturity, fault tolerance and recoverability in different types of use cases.

- Operation faults may endanger human lives or cause great economic or environmental losses, the application must recover without losing any data in any case.
- The software is **either** part of a large integrated real-time system, where all operation faults will cause problems to many other applications and /or thousands of end-users **or** part of a broadly delivered consumer product with extremely high maintenance costs.
- +/- Longer than 2 hour breaks not acceptable, but the system level recovery routines will be adequate. Operation faults never cause remarkable economic or image losses or any danger to human beings.
- + Need for operation not continuous but daily. Time for recovery typically 24 hours and even the worst operation faults will cause losses of not more than hundreds of euros (or dollars).
- ++ Need for operation periodic. Break of a couple of days will cause no harm to the end-user organisation. Typically an independent stand-alone back-office software for a small number of users.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

13. Usability requirements of software

The ease of use and amount of guidance required of the user interface.

- A very big number of different types of end-users all over the world, with different levels of experience of software usage, a high-level of customisation of help facilities required.
- 2-3 different types of users with various skills, requiring automated multi-level help function, software used during interactive customer service.
- +/- A big number of different types of end-users with equal skills, typical simple help function adequate, calm and stressless circumstances for the use of software.
- + Less than 1000 homogenous users at maybe more than one location, literal instructions supported by a very simple help function adequate, no special care for the operability.
- ++ Only a few users, all located at one site, not very frequent use, direct help and support from the developers easy to arrange when needed.
- Don't know.

14. Efficiency requirements of software

Effective use of resources and appropriate performance in every use case and under any reasonable work-load.

- Complex database with millions of data records and transactions per day, thousands of simultaneous end-users with various data requests, almost all response times critical, parallel batch and online processing, operation close to the capacity limits during rush-hours.
- Big database, hundreds of simultaneous end-users, most of response time requirements critical, alternating batch and online processing needs, narrow shifts for batch processing.
- +/- Big database, less than one million data records and less than one hundred simultaneous end-users, response time requirements are flexible, wide shift for batch processing.
- + Average database by volume and structure, straightforward and predictable data requests from few simultaneous end-users.
- ++ Simple and small database, no simultaneous end-users or complex data requests, total number of transactions less than 100 per day.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

15. Maintainability requirements of software

Stability of the environment, target lifetime of the application, criticality of defect diagnostics and test performance.

- Very large strategic software (target lifetime more than 20 years) in a volatile business area with frequent changes of laws, standards and business rules. The maintenance speed is essential, logging and the defect messages must be clear, exact and guiding the developers.
- Large software (target lifetime from 10 to 20 years), frequent changes of laws or standards or business practices. Time allowed to analyse defect messages, change the programs and test them is not more than a few hours.
- +/- Average size tactical software (target lifetime from 5 to 10 years), monthly changes of laws, standards and business practices. Maintenance timing is reasonably flexible, a couple of days rather than hours, an application specific error log needed.
- + Rather small rarely changing software (target lifetime from 2 to 5 years), no application specific diagnostics needed.
- ++ Temporary software (target lifetime less than 2 years) with no connections to changing laws, standards or business practices.
- Don't know.

16. Portability requirements of software

Adaptability and ease of installation in different environments, openness of architecture and structural components.

- Users of the software are located in many kinds of organisations, with various platforms (hardware, browsers, operating systems, middleware, data communication protocols etc), various versions and various upgrading frequencies.
- The software must operate on a couple of different platforms (hardware, browsers, operating systems, middleware, data communication protocols etc) and on several versions of them.
- +/- Every version of the software must run on several versions of a single platform (hardware, browser, operating system, middleware, data communication protocol etc), the upgrading frequencies of the users are rather predictable.
- + The software must run on a single platform (hardware, browser, operating system, middleware, data communication protocol etc), but the use of system level services is limited because the upgrading process is only partially manageable.
- ++ The software must run on a single platform (hardware, browser, operating system, middleware, data communication protocol etc) where the upgrading process is completely manageable (for example most mainframe environments).
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

17. Analysis skills of staff

The analysis skills of the project staff at the beginning of the project.

- No experience in requirements analysis or similar projects.
- Part of the staff (less than 30 %) has experience of analysis and design activities in similar projects.
- +/- 30-70 % of the project staff has experience of analysis work. The project also has an experienced member.
- + Most people have experience of specifications and analysis, the project manager is a professional in analysis work.
- ++ The project staff consists of first-class professionals, strong vision and experience of requirements analysis.
- Don't know.

18. Application knowledge of staff

Knowledge of the application domain by the project team (both the supplier and the customer) at the kick-off moment.

- The application domain experience in the team is very small, less than 6 months on average.
- The application experience is small, some members of the project staff have application experience, from 6 to 12 months on average.
- +/- The team has a quite good experience of working in the application area, 1-3 years on average.
- + The application experience is good both for the supplier and the customer. Business dynamics are known. The experience is 3-6 years on average.
- ++ Both the supplier and the customer representatives know the application area very well, including understanding the overall business. The average experience is more than 6 years.
- Don't know.

19. Tool skills of staff

Average experience of the project team (supplier, customer) in development and documentation tools at the kick-off moment. All important tools must be included here.

- The team has no experience of the necessary tools. The average experience is less than 3 months.
- The tools experience is less than average, some members have experience of some tools, the average is from 3 to 6 months.
- +/- The tools experience is rather good. Some members of the team know the most important tools well. The experience is from 6 to 18 months on average.
- + Most of the team members know the tools needed in the project well. Some members can give support in the use. Experience time from 18 months to 3 years on average.
- ++ The team knows all the tools very well. Support is available for the specific needs of the project. The experience is more than 3 years on average.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

20. Experience of project management

The experience of the project manager and other key staff.

- Project manager has no previous experience of similar types of projects, not even as a member.
- Project manager has experience of smaller similar type of projects, as a member or partially responsible for management.
- +/- Project manager has experience of managing at least one similar project.
- + Project manager is experienced on several similar projects, some success also on difficult projects.
- ++ Project manager is a real professional with experience of managing many kinds of software projects.
- Don't know.

21. Team skills of project team

The ability of the project team members to work effectively together following the best project practices.

- Scattered group of people, no experience of working as a member of a project team, minimal project and collaboration skills.
- Some of the team members have previous experience of similar projects, no experience of working with each other, but some motivation to try exists.
- +/- Most of the team members have experience of similar projects. The commitment to the project goals and the motivation to work together is satisfactory.
- + The project group is active, the team members have experience of working together and all of them are able to share responsibilities rather effectively.
- ++ The team can solve most of the difficult problems together, using innovative and inspiring group work. Superior team spirit.
- Don't know.

EXPERIENCE BENCHMARKING @ DATAMAX

V. Other Classifiers

Project Management Techniques and Methods

Check the ones you used.

- Conceptual analysis
- Cost/benefit analysis
- Effort estimates
- ER-modelling
- Goal analysis
- Load analysis
- Normalisation
- Problem analysis
- Pair testing
- Prototyping
- Reviews
- Risk analysis
- Structural methods
- System testing
- Test planning
- Walkthroughs
- Development model
- Capacity analysis
- Net Profit
- Seer method
- SRM model
- Functional sizing
- Activity modelling
- Other, describe: _____

Customer-Supplier Relationship

The relationship between the customer and the supplier for this project.

- Self made
- Internal supplier
- External supplier
- Multi-supplier project
- Multi-customer project
- Other, describe: _____

EXPERIENCE BENCHMARKING @ DATAMAX

Size of Intended Market

Number of user organisations

- 1
- 2 - 5
- 6 - 50
- over 50

Number of Users

Number of people who will use the software

- 0 – 10
- 11 – 100
- 101 – 1000
- 1001 – 10000
- > 10000

Project Management Tools

Check the ones you used.

- Artemis
- Debugger
- Experience Pro
- Fileaid
- Jsp-tool
- Lotus 1-2-3
- MS Project
- PMW
- TJ Planner
- WP/Word
- Own tool

Other, describe: _____

CASE Tools

Did you use CASE tools? _____

If yes, which ones? _____

Database Management Tools

Did you use DBMS tools? _____

If yes, which ones? _____

EXPERIENCE BENCHMARKING @ DATAMAX

Operating System(s)

Check the ones you used.

- CICS
- DOS
- Guardian Tandem
- IMS
- MVS
- OS 2200
- OS/390
- SVR 4
- Tandem NonStop
- UNIX
- Windows
- Windows NT

Other, describe: _____

User Interface

- Graphical
- Character
- None

Other, describe: _____

Your Comments:

Are there any factors not covered in this workbook which you believe made this project "special"?