Experience MT22 Situation Analysis Model

Purpose of the Experience situation analysis method MT22 is to help to estimate annual maintenance and modification projects. The model consists of 22 standard productivity factors. They are classified into four categories: Organisation (6 factors), Process (5 factors), Product (6 factors) and People (5 factors). Each factor in each category has five alternative values. The basic idea in rating is that “the better the circumstances for the maintenance are, the more positive rating the factor gets”.

“++” = Excellent situation, circumstances much better than in average case

“+” = Good situation, circumstances better than in average case

“+/-“ = Normal situation in the productivity point of view

“-“ = Bad situation, circumstances worse than in average case

“--“ = Very bad situation, circumstances much worse than in average case

Rating of each factor is weighted based on experience data. The ideal or target weights should be 1.10, 1.05, 1.00, 0.95 and 0.90 (from - - to ++) and they should be distributed normally, 5 - 20 - 50 - 20 - 5 %.

Categories and names of M22 Productivity Factors

Category	**Name of productivity factor**
Organisation	Release and version policy
Organisation	Resource availability for future needs
Organisation	Contracting procedure
Organisation	Number of stakeholders
Organisation	Priority setting and control of changes
Organisation	Organisational culture
Process	**Source code edition methods and tools**
Process	**Testing methods and tools**
Process	**Documentation methods and tools**
Process	**Communication methods and tools**
Process	**Roll-out methods and tools**
Product	Functionality requirements
Product	Reliability requirements
Product	Usability requirements
Product	Efficiency requirements
Product	Maintainability requirements
Product	Portability requirements
People	**Development environment skills of staff**
People	**Application knowledge of staff**
People	**Networking skills of staff**
People	**Motivation and responsibility of staff**
People	**Team atmosphere**

1. Organisation factors 1.1 – 1.6

1.1 **Release and version policy;** The clarity, formality, internal integrity and long-term planning of future releases and versions.

-- Many different customer specific versions and delivery packages, which are built case by case after required modifications.

- Many customer specific versions and release packages, and their installation and deployment is allocated to end-users.

+/- Some customer specific versions and releases of the same delivery.

+ All customers get the same standard delivery. New versions are released according to future needs.

++ All customers get the same standard delivery. Future releases are planned and agreed for the foreseen future.

1.2 **Resource availability for future needs**; adequacy of resources and systematic allocation of staff, hardware, software, work space and required skills for the planned maintenance period.

-- Organisation has no defined rules and practices in resource management. Applications have no responsible person. Continuous lack of resources.

- Organisation has mechanism for workload management, and it is at least partially in use. Each application has responsible person, but he/she has many other responsibilities. Availability of resources is uncertain.

+/- Organisation has defined mechanism to manage critical resources. Each application has responsible person(s). Some delays to get other resources.

+ Organisation has well-defined mechanism to manage all resources, and it is followed largely. Responsibilities are fully allocated to suitable person(s) and also required back up resources are nominated. Other resources are available on request with short notice.

++ Organisation has well-defined mechanism to manage all resources, and it is followed fully. All required responsibilities and back up resources are nominated and their availability is well ensured. Also other resources are available on request.

1.3 **Contracting procedure;** Consistency, completeness and granularity of maintenance contract to define each service type and/or service transaction, mutual responsibilities, level of services, acceptance criteria of deliveries and other required contract conditions.

-- Maintenance service is not based on any contract or other documented practice.

- Maintenance service is performed according to continuous framework agreement, but separate services and deliveries are not identified.

+/- Maintenance service is based on continuous framework agreement, and each service transaction is recorded by supplier and accepted at least orally by customer/end user.

+ Maintenance service is based on separate service agreements and each service delivery is based on mutually accepted documents.

++ Each service type is based on documented mutual agreement, and is an element of continuous framework agreement. Each delivery is based on mutually approved specification document.

1.4 **Number of stakeholders;** Number of people and/or organisations involved in management and decision making of maintenance service and deliveries.

-- Number of people and organisations involved in implementation and decision making of change requests is high (both more than 5).

- Either the number of people or organisations involved in implementation and decision making of change requests is high (either # of people or organisations more than 5).

+/- Number of people and organisations involved in implementation and decision making of change requests is typical/average (both 2-4).

+ Either the number of people or organisations involved in implementation and decision making of change requests is low (1-2) and the other is not high (not more than 5).

++ Number of people and organisations involved in implementation and decision making of change requests is low (both 2 or less).

1.5 **Priority setting and control of changes;** Classification and analysis of change requests by defined criteria (for example criticality, urgency and cost), to prioritise change requests and decide on required actions of both parties.

-- No agreed classification for errors, failures and change requests.

- Only application specific error classification is in use.

+/- Organisation wide error and failure recording and classification is in use and it is used to prioritise fixing actions.

+ Organisation has classified each application by business criticality and classifies also each error and failure respectively. Each change request is analysed by benefit/cost method.

++ All applications and error and failure types have widely known criticality classification and consistent benefit/cost analysis method. All responsible persons (operators etc.) are fully aware of all problem situations and required actions.

1.6 **Organisational culture;** Common attitudes among staff and appreciation of maintenance at company level, appropriate awarding mechanism and other cultural factors.

-- Organisation and people are enthusiastic of new technologies and projects only. New development projects are highly appreciated, maintenance “just must”. No visibility for maintenance work, no awarding mechanism for maintenance projects and services.

- Importance of maintenance is known, but not shown. No communication and awarding mechanism for maintenance.

+/- Organisation values maintenance but does not motivate people in maintenance work in any means. Maintenance is mentioned in top management presentations and is part of company-wide measurement program.

+ Maintenance has good image in company as a key long-term success factor and profit maker. Maintenance is a profession, and is part of recruiting campaigns.

++ Maintenance has good image in company and has strong motivation and commitment among top management and staff. People want maintenance responsibilities and activities. Maintenance is measured at organisational, delivery and individual levels and is part of awarding mechanism.

2. Process factors 2.1 – 2.5

2.1 **Source code edition methods and tools**; The level and impact of code editors, translation tools, code libraries and code integrity tools and procedures.

-- Development environment and tools are not in proper use and widely known. Several hardware platforms.

- Development environment and tools are in moderate use, but are immature and new versions are needed frequently. Some guidelines and standards are in partial use.

+/- Development environment and tools are in common use. Guidelines, procedures and standards are created, but only in partial use.

+ Development environment and tools are well known and in common use. Guidelines, procedures and standards are in use and easily accessible.

++ Environments and tools are an integrated set, and automate major parts of manual tasks. Simple, well-known development environment and only one hardware platform.

2.2 **Testing methods and tools**; Level and impact of tools and procedures to manage test cases and materials, test activities, regression tests and test results.

-- No testing practices and standards. Test materials are derived separately each time when required.

- Testing activities and standards exist, but test case derivation and reuse is difficult. All data is file based, only manual handling of files and data.

+/- Testing is well performed and largely supported by standards. Test data is managed with appropriate tools and/or scripts.

+ One test material package, which can be modified for different test situations. Testing process and appropriate tools are well documented and in proper use.

++ Each application/software component has well-defined test suite (scripts and materials) for all defined testing phases. Regression testing is tool-supported, where appropriate.

2.3 **Documentation methods and tools;** Level and impact of tools and procedures to create, manage and distribute required application documents for maintenance staff and end users.

-- No common procedure and widely used professional practices for documentation.

- No common guidelines and procedures for any documentation, only some version and change control in use.

+/- Good documentation of each application, change request and error/failure. Follow-up of documented changes, errors and failures is in use.

+ Application documents are well managed, controlled and maintained. Mostly manual documentation.

++ Well-defined process for documentation of each application. Documentation is tool based and in wide, firms use.

2.4 **Communication mechanisms**; Level and impact of methods, tools and procedures to record, communicate and handle change requests, errors and failures.

-- No defined approach for communicating. Required information is distributed for all potential parties to avoid “communication gap”. Many kinds of media are in use.

- Communication mechanism is defined, but only in partial use. Some guidelines are available.

+/- Communication mechanism is documented, and in proper use. It is not integral part of maintenance process.

+ Communication is well integrated with maintenance activities and process. No tools, but some templates and distribution lists are in use.

++ Multiple tools for communication are in proper use and well aligned with actual work processes. Templates support major part of communication.

2.5 **Roll-out methods and tools;** Level and impact of tools and procedures to roll-out modified programs/applications and related data to operation environment.

-- No defined approach for rollouts. Deployment is work intensive and depends on key staff.

- Some documents about rollouts are made and in use. Some separate tools in use.

+/- Roll-out is a well-defined process and it is followed largely. One dominant tool to perform roll-out and record roll-out status. No easy traceability and version status of rollouts.

+ Well-established work process for roll-out. Good version control and traceability.

++ Roll-out and version control has already long history in organisation and is fully automated.

3. Product factors 3.1 – 3.6

3.1 **Functionality requirements;** Variety and complexity of the requirements and business rules, level of interfaces.

-- Virginal and complex application area, security critical big (thousands of FP’s) multi-tier system for various, multi-cultural users. Many authorisation levels for users. Some complex, algorithmic functions.

- Various user groups and access levels to applications and data. Many interfaces with other systems. Some business rules require special application knowledge from developers and testers.

+/- Some user groups with slightly different access control. Mostly simple business rules.

+ Only a couple of user groups, only some interfaces with other systems. All business rules are relatively simple.

++ Only one user group, all have same access control. No interfaces with other systems. Functionality is simple data movement to and from user (screens, reports).

3.2 **Reliability requirements;** Severity of failures and impact of failures to users and operation.

-- Operation faults may endanger human lives or cause great economic or environmental losses, the application must recover without losing any data in any case.

- Failures can cause major economic loss and image suffering, can lead to negative news in mass media.

+/- Faulty operation can cause harm for some hundred users, can reflect negatively in operation of 2-3 other applications.

+ Failures can cause harm for some tens of users, but they can tolerate short operation breaks even daily. Some impacts in max one other application.

++ Failure has impacts only in some users. Maximally weekly or monthly operation period, error can be fixed without operational losses. No impacts on any other applications.

3.3 **Usability requirements;** Number of users, support for various skill levels of users, continuous operation, special requirements to attract users.

--- A very big number of different types of end-users all over the world, with different levels of experience at software usage, a high-level customisation and help facilities required. 24 h/day, 7 days/week operation requirement.

- 2-3 different types of users with various skills and languages, requiring automated multi-level help function, the use of software during interactive customer service. 24 h/day operation requirement.

+/- Limited number of regular users, who can be trained in advance. Mostly in back-office use, sometimes in direct customer service. Max 20 h/day operation.

+ Application for small number of users. Only in back-office functions. Operation in working hours is required.

++ Only few expert users or one team, all located at one site, not very frequent use.

3.4 **Efficiency requirements;** Requirements for response and transaction processing time, differences in operational and computer load, transaction and data volumes.

-- Very big volume of real-time transactions, big differences in operation load, need for simultaneous on-line and batch processing. Millions of records in database, many kinds of non-predictable inquiry needs.

- Hundreds of simultaneous end-users in multiple sites, most of response time requirements critical, queuing in transaction processing causes operational loss for services.

+/- Max one hundred simultaneous end-users. Response time requirements are flexible but critical for work efficiency, mostly only predefined inquiry needs.

+ Simple database, 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 not more than tens per day.

3.5 **Maintainability requirements;** Stability of the environment, standardised code and component structures, clarity of architecture, pressure for changes.

-- Very large strategic (target lifetime more than 20 years) software at a volatile business area with frequent changes of laws and standards and business rules. Also the maintenance speed is essential, logging and the defect messages must be clear, exact and instructive for developers.

- Large software (target lifetime from 10 to 20 years), frequent changes of laws or standards or business rules. Time to analyse defect messages, change the programs and test them is always some hours but not more.

+/- Average size tactical (target lifetime from 5 to 10 years) software, monthly changes of laws, standards and business rules. 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 intention to enhance for new requirements.

3.6 **Portability requirements;** Adaptability and installability to different environments, openness of architecture and structural components, volatility of platforms and environments.

-- Users of the software are located in many kind 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 many 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 certain 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 certain platform (hardware, browser, operating system, middleware, data communication protocol etc), for which the software is tested. Only one “latest version” of software is required. Some customers or user groups may use older versions, but they don’t need to be interoperable with new version.

++ The software must run only on a certain platform (hardware, browser, operating system, middleware, data communication protocol etc) which upgrading process is completely manageable (for example most of the mainframe environments). Several tens of similar applications are running on the same platform.

4. People factors 4.1 – 4.5

4.1 **Development environment skills of staff;** Experience and knowledge of maintenance staff in development environment, tools and platforms (design, implementation, testing, version control, operation, documentation, communication)

-- Development environment and tools are new for the whole maintenance staff. The average experience time is less than 3 months. Special expertise is difficult to get. Training needs are not satisfied.

- At least one responsible person has reasonable knowledge of environments (3 months – 2 years). Special knowledge is difficult to get. Training is partially available.

+/- At least one of some responsible persons have good knowledge of environments (several years). Average experience is 1 –3 years. Special knowledge is largely available on request. Training is available on essential tools.

+ All responsible persons know well the environments and tools (2 – 5 years). Some persons can give hands-on support in tools. Training is available on all tools.

++ The whole maintenance staff knows all the tools very well (>5 years experience). Support available for the specific needs of the project. No need for training.

4.2 **Application knowledge of staff**; Knowledge of the maintenance staff in the applications and interfacing systems (both the supplier and the customer).

-- The business area knowledge of maintenance staff is very small, less than 12 months. No expertise on interfacing systems.

- The application experience is small on vendor side, and software knowledge is small on customer side. Maintenance staff has no special knowledge on interfacing systems.

+/- Maintenance staff has quite good experience of the business area and application domain, 1-3 years in average. At least some people have good overall understanding of the application portfolio.

+ The business area and application domain experience is good both on the supplier and the customer sides. The experience is 3-6 years in average, some have >5 years experience.

++ Both the supplier and the customer representatives know the business area and application domain very well (in average >5 years), including the understanding of the business as total. Good understanding of application portfolio among the whole maintenance staff.

4.3 **Networking skills of staff;** Level of teambuilding and networking among maintenance staff, ability to cooperate with partners.

-- Maintenance staff consists of new people, no mutual working history and experiences. Responsible persons have no common language. No connection with external stakeholders.

- Part of maintenance staff has common working history, max 2 years. Management and experts have mutual communication and understanding problems. Ad hoc connections with stakeholders.

+/- Maintenance staff has some year's common working history (2 – 5 years). Mutual communication is open and works quite well. Cooperation with stakeholders is done when required.

+ Maintenance staff has long common working history (>5 years). No mutual communication problems between management and experts. Regular cooperation with stakeholders.

++ Maintenance staff has very long common working history (>10 years). No communication problems between management and experts. Stable and continuous cooperation with stakeholders, even when responsibilities are allocated to new staff.

4.4 **Motivation and responsibility of staff;** Personal motivation to develop application and customer business

-- Maintenance staff has no interest to develop application. Maintenance is considered as mandatory extra duty, which should be avoided. Easy to transfer responsibility to other staff members.

- Maintenance staff members are not interested to develop application, expect some limited responsibilities that are explicitly allocated to them.

+/- Maintenance staff members are performing maintenance activities according to plan and take the development responsibility.

+ Maintenance staff members are interested to develop application and take personal responsibility over the whole application area, as defined.

++ Maintenance staff is interested to develop customer's business, like introduction of new technology, competitive position of applications and new changes in interfacing systems. Real responsibilities are far over the minimal requirements defined in maintenance contract.

4.5 **Team atmosphere;** Influence on working conditions, self-learning, professional career opportunities.

-- Maintenance staff feels that their work effort is highly underappreciated. Continuous lack of resources. No influence on daily work and working conditions. Unfair or unknown feedback on work.

- Maintenance staff feels that their work is underappreciated, and leads to at least temporary resource conflicts and inadequate training. Only some influence on daily work at individual level, weak feedback on work results.

+/- Maintenance staff feels that their work is moderately appreciated. Resourcing and training are quite adequate. Mostly good influence on daily work, sometimes resource conflict with continuous responsibilities and project duties.

+ Maintenance work and results are well appreciated. Resourcing and training are adequate. Each individual has good influence on daily working arrangements. Good feedback from work, fair awarding.

++ Excellent feelings about maintenance work among the whole staff and management. Resourcing and training are adequate. Good knowledge on all feedback from management and customer, awarding is fair. Full responsibility and self-control at individual level on personal working conditions and satisfaction of new professional requirements.