Annex to the GTC for the development of software according to agile methods ("Agile Software Development")

In addition to the GTC the following provisions shall apply for the development of software in agile software development projects. Unless otherwise defined herein, capitalized terms used herein shall have the meaning ascribed to them in the GTC.

1. Definitions

Agile Software Development is a framework within which people can tackle complex adaptive tasks. It consists of agile teams and the roles belonging to them (Product Owner, Agile Master, Development Team,...), events (Sprint, Sprint Planning, Daily Agile Meeting, Sprint Review, Sprint Retrospective,...).), artifacts (Product Backlog, Sprint Backlog, Increment,...) and rules.

An ***Agile Team*** consists of the Product Owner, the Development Team and the Agile Master. It is self-organizing and decides for itself how best to do its job.

The ***Product Owner*** is responsible for maximizing the value of the product, and thus for managing development performance, gradually formulating and substantiating the business and functional expectations of a product to the development team. The Product Owner is the only person responsible for the management, in particular the mandatory creation and updating, of the Product Backlog. He can be supported in managing the product backlog, but always remains accountable.

The ***Agile Master*** is responsible for promoting and supporting agile software development. Agile Masters do this by helping all participants to correctly understand and use the theory, practices, rules and values of the Agile Framework, and above all to organize the sprints and their events. The Agile Master is responsible for supporting the Agile team in achieving its goals and eliminating impediments. This includes managing the impact of the environment on the Agile team and coordinating coordination with the environment.

The ***Agile development team*** is responsible for the management, organization and execution of all development work required to create an increment. It is interdisciplinary and has all the skills necessary to do the job. Development teams are structured and empowered by the organisation to organise and manage their own work.

The ***sprint*** is a fixed period of time [time box], usually a maximum of one month in length, within which a finished ["done"], usable and potentially deliverable increment is produced. All sprints within a development project should have the same duration. The new sprint starts immediately after the completion of the previous sprint. A Sprint includes and includes all agile activities and events, especially Sprint Planning, Daily Meetings, Development, Sprint Review and Sprint Retrospective.

Sprint Planning is a temporary event, usually 8 hours or less [Time Box], to start a sprint. It is used by the agile team to evaluate, plan and transfer the work from the product backlog that needs to be done next to the sprint backlog.

The ***Sprint Retrospective*** is a time-limited event of usually 3 hours or less [Time Box] to stop a sprint. It is used by the Agile team to review the past sprint and plan improvements for the next sprint.

The ***Sprint Review*** is a time-limited event of usually 4 hours or less [Time Box] to review the to complete the development work on a sprint. It serves the agile team and stakeholders to inspect the increment resulting from the sprint, assess the impact of the work performed on overall progress, and verify the successful implementation of the relevant requirements. If necessary, the product backlog can be adapted on the basis of the findings of the Sprint Review.

The ***product backlog*** is an ordered list of everything known to be included in the product. It serves as the sole source of requirements for all changes to the product. The Product Owner is responsible for the Product Backlog, its contents, the access to it and the order of the entries. A product backlog is never complete. During its first

development steps, it shows the initially known and best understood requirements. The product backlog evolves with the product and its use. It is dynamic; it constantly adapts itself to make clear for the product what it needs to be appropriate to its task, to compete and to offer the necessary benefits. The entries in the product backlog are recorded in the form of a user story.

Product backlog refinement is a continuous process in which the product owner and the development team jointly detail, review, estimate, prioritize and revise product backlog entries. The product owner can also update the entries in the product backlog or have them updated at any time outside the refinement.

The ***sprint backlog*** is the set of product backlog entries selected for the sprint that is necessary to deliver the increment and achieve the sprint goal [Tasks]. The Sprint Backlog is a forecast by the development team of what functionality will be included in the next increment and the work required to deliver that functionality in a finished ["Done"] increment. The Sprint Backlog makes visible all the work the development team deems necessary to achieve the Sprint goal.

The ***increment*** is the result of all product backlog entries completed in a sprint and thus supplements or extends the result of the increments of all previous sprints. At the end of a sprint, the new increment must be ready ["Done"]; that is, it must be in an inspectable and usable state. Increment is a step towards a vision or a goal. It must also be in operational condition if the product owner does not currently want to deliver it. The sum of all increments ultimately forms the developed product.

Impediments are obstacles for the Agile team that occur during a sprint and hinder the team in fulfilling its task. Impediments are eliminated by the Agile Master.

A ***user story*** is a requirement formulated in everyday language. It is deliberately formulated briefly and usually consists of only a few sentences, including the desired goal and certain acceptance criteria. The product owner writes the user stories and adds them to the product backlog.

2. Common Understanding of the Processing of Individual Orders / Basic Rules

2.1. Independent Service Provision / Subcontractor

2.1.1 The Contractor shall provide the contractual services independently and under its own responsibility.

2.1.2 The Contractor is generally free to choose the place of performance for the performance of his services. However, if the project requires the services to be carried out in part on the premises of the Customer or of a third party, the Contractor shall be prepared to carry out the services in question on those premises and the place of performance shall be agreed between the contracting parties taking into account the requirements of the project.

2.1.3 The Contractor shall have sole authority to issue instructions to its own employees and any subcontractors employed by it. He is free to organize the provision of services and to schedule the time of his activity. However, to the extent that the project so requires, it shall consult with other participants in the project in order to meet agreed deadlines.

2.1.4 In the event of the deployment of employees, vicarious agents and subcontractors, the Contractor warrants that all necessary official approvals (such as work permits, residence permits) have been obtained. The Contractor shall indemnify the customer against all legal consequences resulting from non-compliance with this requirement.

2.1.5 The Contractor and its subcontractors shall be responsible for the fulfilment of all employer obligations, in particular with regard to the payment of salaries and social security contributions. The Customer shall not be liable for the payment - in particular - of salaries, daily rates, taxes, social security contributions and insurance contributions for staff employed by the Contractor or its subcontractors. Neither a framework contract nor an individual contract shall create an employment contract between the

Customer and any person employed by the Contractor or a subcontractor.

2.1.6 The assignment of a subcontractor (including external consultants and freelancers) requires the written consent of the Customer, which can be denied without giving reasons. Subcontractors requested by the Contractor must be stated in the offer with the name of the consultant and the company data. Affiliated companies of the Contractor are also subcontractors within the meaning of this Clause.

2.2. Participation of A1/ Agile Teams in the Project Organization

According to the agile software development method, the employees of the Customer and the Contractor work together in Agile Teams. Usually the project organization consists of a product owner (provided by the Customer), an agile master (provided by the Customer), and the developer team (provided by the Contractor, may include employees of the customer, or other vendors - "mixed teams"). Thus the participation of A1 is necessary wherever A1 employees have roles and responsibilities related to the Agile Team.

That includes:

- Planning of all appointments and sprints together with the Contractor
- Prioritization of requirements and creation of backlog for sprints
- Holding regular daily meetings, sprint meetings and sprint reviews

Within the framework of the provision of services from this service module, A1 as the Customer shall fulfil the following obligations to cooperate:

- Specification of the requirements for the software to be created, and transmission of these in the form of clearly defined basic functionalities to the contractor (Requirements),
- Nomination of employees as contact persons for voting, regular project meetings, sprints, escalations etc.
- Introduction to the Customer's development and commissioning processes,
- Provision of relevant documentation, guidelines and guidelines for processes, architectures and applications,
- Preparation and timely provision of information, data, hardware, operating systems, standard software used, organization charts and other components necessary for the program production,
- Acceptance and testing of the individual deliverables such as concepts, basic software, end software, protocols etc. and rapid feedback to the contractor (in writing).

Disturbances and impediments must be channelled through the Agile Master by the Contractor or the persons deployed by him and must not be independently clarified with other employees in the Customer's organisation. Other queries from the Contractor or the persons deployed by the Contractor, e.g. to the Product Owner during a sprint, must also be coordinated via the Agile Master.

3. RESULTS (DELIVERABLES)

The following results are typically delivered by this power module:

- Software concept and design
- Customized, fully functional software and customizing (adaptations, enhancements)
- Source code and descriptive and explanatory documentation, the minimum scope of which shall be such as to enable an understanding of the structure and operation of the program after a reasonable period of familiarisation.
- Documentation of the delivered software functionality (operating manual, user manual, installation instructions)
- Training, user training

4. ACCEPTANCE

After each sprint, the deliverables *(user stories)* implemented in this sprint are accepted. The contractor declares to the client that the user stories jointly agreed for the sprint are ready and correspond to the acceptance criteria agreed at the beginning of the sprint *(Definition of Done).*

In principle, a sprint is only accepted in its entirety - i.e. if all agreed user stories meet all agreed criteria. Sprint dates are agreed between Product Owner, Agile Master and Agile Team and can be included in the offer.

The final acceptance of the entire product (Final Acceptance) shall take place after completion of the final version of the fully functional software ready for rollout or after installation of the program on the hardware of the Customer and initial instruction.

Acceptance shall take place at least in the following steps:

- after installation of the program, the Contractor shall demonstrate the existence of the agreed characteristics and essential functionalities by means of appropriate acceptance tests; and
- after successful acceptance tests, the Contractor and Customer jointly draw up an acceptance protocol in which any defects must also be recorded, and
- a written declaration of acceptance will be issued by the person responsible on behalf of A1.

5. KEY PERFORMANCE INDICATORS (KPIS)

While in the waterfall method the KPIs only allow a quantitative evaluation of project success and efficiency after completion of a project, in agile software development two categories of KPIs are meaningful. In addition to the KPIs that are determined after the project has been completed, agile software development also includes informative KPIs that provide information about the current status of an agile project or project progress during the project period.

5.1. Target values and measurement during the project period

5.1.1. Sprint Burndown (project progress; productivity)

Definition and target value	• Story Points/Sprint; agreed individually for each project • Sprint Burndown is the degree to which the backlog tasks in an ongoing project are dismantled or completed. It thus shows the current progress of a project. • Sprint Burndown is directly related to velocity (see 3.1.2) and is measured in Story Points/Sprint like this.
method of measurement	The basis for the measurement is the Sprint Burndown Chart. It shows the task completion over time.

5.1.2. Team Velocity (project progress; productivity)

target value	• Story Points/Sprint; agreed individually for each project. • In the case of new (not yet established) teams/methods, it should also be agreed by which value the velocity should increase during the calculation period. Velocity is supposed to increase over time, as the AGILE team has become well-rehearsed at some point and has gained know-how and experience.
method of measurement	Number of Story Points that a team has implemented in a sprint. The measurement is based on the Sprint Burndown Chart and the data from the A1 tracking systems.

5.1.3. Scope Creep (In-Budget Delivery)

definition	• User Stories/Sprint • Number of additional user stories/sprint added to the original scope. Scope Creep thus

	corresponds to a Scope extension during the duration of the project.
method of measurement	Difference between the actual number of user stories in the sprint and the number of user stories originally planned in the scope.

5.1.4. quality

definition	• Defects/User Story • Max. Number of defects is individually agreed per stage (INT/PROD,..)
method of measurement	Number of existing defects/user stories per stage. Categorization of defects follows classification and evaluation in the A1 tracking system.

5.1.5. Automated test coverage *(quality)*

Definition and target value	• Percentage of code passed in total program code when executing tests • Target value: min. 80%
method of measurement	• Fully automated tests • Calculation (%) = (number of code lines run through/total number of code lines)*100

5.2. Target values and measurement after project completion

5.2.1. In-Budget Delivery

target value	€ 0,-- Budget overrun
method of measurement	Deliverables implemented in relation to (still) available planned budget

5.2.2. unit cost

target value	Fixed EUR value per project
method of measurement	EUR/Story Point

5.2.3. Delivered Defect Density to UAT

target value	Maximum value agreed individually
method of measurement	Number of errors per Epic/EUR/FP/Jira req.

5.2.4. Resolution Time for Blocker

target value	≥90% of all blockers in UAT solved within 4 hrs.

method of measurement	Number of blockers solved per time unit

5.2.5. Iterations to solve INT defects

target value	>90% are dissolved in the first iteration.
method of measurement	Number of attempts that result in a ticket being resolved