Capturing Agile Requirements by Example (CARE)

Glossary

Version: 1.0

Released: September 2022

Copyright Notice

This document may be copied in its entirety, or extracts made, if the source is acknowledged.

All Agile United syllabi and linked documents (including this document) are copyright of Agile United (hereafter referred to as AU).

The material authors and international contributing experts involved in the creation of the Agile United resources hereby transfer the copyright to AU. The material authors, international contributing experts and AU have agreed to the following conditions of use:

- Any individual or training company may use this glossary as the basis for a training course if AU and the authors are acknowledged as the copyright owner and the source respectively of the glossary, and they have been officially recognized by AU. More regarding recognition is available via: https://www.agile-united.com/recognition
- Any individual or group of individuals may use this glossary as the basis for articles, books, or other derivative writings if AU and the material authors are acknowledged as the copyright owner and the source respectively of the glossary.

Thank you to the main author

- Kaspar van Dam

Thank you to the co-authors

- Piet de Roo
- Patrick Duisters
- Mark de Munnik

Note about references in this document

Throughout this Glossary there are references to websites or other sources. All terms and their descriptions are written down using information from mentioned sources at the moment of writing. Terminology may have been taken directly from these sources or it may have been based on information presented at these sources. If any copyright may have been infringed this has been unintentional. In case of an infringement, please notify the author(s) so it can be corrected.

Within descriptions other terms (or abbreviations) may be used or referred, which are also described within the Glossary. The referred terms are underlined.

Revision History

Version	**Date**	**Remarks**
1.0	September 2022	Initial Public Release

Terms, abbreviations, and descriptions

Term / abbreviation	**Description and reference**
Acceptance criteria	Criteria that a component or system must satisfy in order to be accepted by a user, customer or other authorized entity. [ISO 24765, ISTQB Glossary]
Acceptance testing	Testing conducted to determine whether a system satisfies its acceptance criteria and to enable the customer to determine whether to accept the system. [ISO 24765]
Acceptance Test Driven Development (ATDD)	Acceptance Test Driven Development (ATDD) involves team members with different perspectives (customer, development, testing) collaborating to write acceptance tests in advance of implementing the corresponding functionality. [https://www.agilealliance.org/glossary/atdd/]
Actor(s)	People (or groups thereof) who can impact the desired outcome. Including: 1. Those who can (help) produce the desired effect; 2. Those who can get in the way of achieving the goal; 3. The consumers who are expected to benefit from the product; 4. People (within the organization) that will be impacted by it. [https://www.keepsolid.com/goals/glossary/impact-mapping]
Agile manifesto	A statement on the values that underpin Agile software development. The values are: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation, responding to change over following a plan. [ISTQB Glossary]
Agile requirements	Agile requirements may be defined upfront, but they can also be further detailed during agile development as part of a user story, while the developer and end user/customer work closely together and adapt the requirement to what is really needed. As such, at least when the user story is done, the product is developed, and the requirement is clear. (See also: Requirements) [CARE Syllabus]
Agile (software) development	Software development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries, in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback. [ISO 24765]

Term / abbreviation	Description and reference
Agile testing	A testing practice that follows the rules and principles of agile software development. Unlike the Waterfall method, Agile Testing can begin at the start of the project with continuous integration between development and testing. Agile Testing methodology is not sequential (in the sense it's executed only after coding phase) but continuous. [https://www.guru99.com/agile-testing-a-beginner-s-guide.html]
Automate validation	Automating the tests that are meant to answer the question if the software delivers what the end-user or business really needs. They are about helping to ensure that the software fulfills the desired use in the appropriate environment. [CARE Syllabus]
Behaviour Driven Development (BDD)	BDD is a second-generation, outside-in, pull-based, multiple-stakeholder, multiple-scale, high-automation, Agile methodology. It describes a cycle of interactions with well-defined outputs, resulting in the delivery of working, tested software that matters. [Dan North @ "Agile specifications, BDD and Testing eXchange" in November 2009 in London]
Business goals	An objective or intended outcome from a business perspective. Mostly related to sales, profit. In the context of CARE: What the business wants describing certain business needs from an end-user point of view. When these goals are split up in business rules, describing a single piece of functionality which a software product can perform, then these business rules can be used as key examples within Specification by Example. (See also: Goal) [CARE Syllabus]
(Business) process modelling	The activity of representing some elements of a process , device, or concept. [ISO 24765]
Business-faced	From the business perspective, often related to validation. [CARE Syllabus]
Capturing Agile Requirements by Example (CARE)	A mindset, a way of working and a tool, combining agility, communication, and collaboration, based on the combined principles of Behaviour Driven Development (BDD), Impact Mapping, Event Storming, Example Mapping and Specification by Example. It provides a means to practically implement these things and to help create software an end-user really cares about. [CARE Syllabus]
Command	Expression that can be input to a computer system to initiate an action or affect the execution of a computer program [ISO 24765]

Term / abbreviation	**Description and reference**
Context – Action – Expected Outcome	Recommended syntax for the examples used within Example Mapping. [CARE Syllabus]
Cucumber (tool)	Tooling used for BDD. Cucumber supports the Gherkin language and can help creating living documentation and provides a framework in which Gherkin can be used to control test automation. Cucumber was originally created in Ruby, but there's also a Java version available. (See also: SpecFlow)
Deliberate discovery	Investing effort in firstly discovering which aspects of delivery we are most critically ignorant of (i.e., both where we are ignorant and where that ignorance is hampering throughput), and further to invest in reducing that ignorance, this to relieve the constraints and allowing to proceed. The things we are still ignorant about can be divided into known unknowns and unknown unknowns. Things we realize we're still ignorant about and things we don't even yet realize we don't yet know. [https://dannorth.net/2010/08/30/introducing-deliberate-discovery/]
Deliverable	Any unique and verifiable product, result, or capability to perform a service that must be produced to complete a process, phase, or project [ISO 24765]
Delivery mapping	Mapping your delivery (strategy) in advance, especially focusing on skills (skills liquidity) and how to improve the teams' skills using Kaizen. [https://speakerdeck.com/tastapod/introducing-delivery-mapping]
Developer-Faced	From the development business perspective, often related to verification. [CARE Syllabus]
Development test(ing)	1. formal or informal testing conducted during the development of a system or component, usually in the development environment by the developer. 2. testing conducted to establish whether a new software product or software-based system (or components of it) satisfies its criteria. [ISO 24765]
Domain-Driven Design (DDD)	Domain-driven design (DDD) is an approach to developing software for complex needs by deeply connecting the implementation to an evolving model of the core business concepts. Its premise is: - Place the project's primary focus on the core domain and

Term / abbreviation	Description and reference
	domain logic - Base complex designs on a model - Initiate a creative collaboration between technical and domain experts to iteratively cut ever closer to the conceptual heart of the problem. [http://dddcommunity.org/learning-ddd/what_is_ddd/]
Event	Occurrence of a particular set of circumstances, external or internal stimulus used for synchronization purposes, or a change detectable by the subject software. [ISO 24765]
Event Storming	Event Storming is a flexible workshop format for collaborative exploration of complex business domains using cross-discipline conversations between stakeholders with different backgrounds, lowering specialization boundaries. [https://www.eventstorming.com/]
Example Mapping	Interactive workshop for collaboratively discovering and capturing stories and divide them into rules. In turn these rules are divided into unique, concrete examples. Any possible questions or assumptions that arise during this workshop are also written down (instead of having endless debates about them during the session). [https://cucumber.io/blog/example-mapping-introduction/]
Executable Specification	Specification written down using a structured and common domain-specific language using the 'Gherkin'- or 'Given-When-Then'-format). While it should be natural language with abstraction close to the business domain it should also be structured enough to be (automatically) turned into executable (automated) tests. [https://cucumber.io/blog/hiptest/what-are-executable-specifications/]
Extreme Programming (XP)	Extreme Programming empowers your development team to confidently respond to changing customer requirements, even late in the life cycle. It emphasizes teamwork. Managers, customers, and developers are all equal partners in a collaborative team. Extreme Programming implements a simple, yet effective environment enabling teams to become highly productive. The team self-organizes around the problem to solve it as efficiently as possible. Extreme Programming improves a software project in five essential ways: communication, simplicity, feedback, respect, and courage. [http://www.extremeprogramming.org/]
Feature (file)	A Feature within BDD is used to specify certain required functionality using the Gherkin language. The form used may differ per software development team. Usually a Feature consists of a title, a feature story describing the functionality (often in a

Term / abbreviation	Description and reference
	similar form as a user story: 'As a ..., I want..., So that...'), business goal, key examples and/or business rules and scenario's which specify the required functionality with examples.
Gherkin	Gherkin is a business readable, domain specific language that enables lets you describe software's behaviour without detailing how that behaviour is implemented. It serves two purposes: Documentation and automated checks. The most important elements in Gherkin are Scenario and the Given-When-Then structure.
Given-When-Then	Syntax in Gherkin specifying the intended behavior of a single piece of functionality with an example: Given <the following prerequisites>, When <the end user performs this behavior>, Then <the system should have responded with that behavior>.
Goal (The)	Objective or Intended outcome. Part of The Goal, as introduced by Eliyahu M. Goldratt in the book of the same name which discusses how the goal should always be the central point of attention: what is it you want to achieve? What's standing in the way to achieve that goal faster/cheaper/more efficient and/or more effective? It isn't effective to improve parts of the process if it doesn't mean an improvement to achieving the goal. Instead, you should always focus on certain bottle necks that are in the way of achieving the goal. These bottlenecks always exist, which is also referred to as the Theory of Constraints. [Goldratt, The Goal (1984)]
Impact Map	Result of an Impact Mapping sessions: A visualization of scope and underlying assumptions, explaining how deliverables connect to user needs, and communicate how user outcomes relate to higher level organizational goals. Created collaboratively by (senior) technical and businesspeople by answering the questions Why? Who? How? and What?, often in the form of a mind map. See also: Impact Mapping [CARE Syllabus]
Impact Mapping	A strategic planning technique, collaboratively identifying and visualizing of scope and underlying assumptions, explaining how deliverables connect to user needs, and communicate how user outcomes relate to higher level organizational goals. [CARE Syllabus]
Incremental (software) development	Software development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall software product [ISO 24765]

Term / abbreviation	Description and reference
Iterative software development / iterative life cycle	A project life cycle where the project scope is generally determined early in the project lifecycle, but time and cost estimates are routinely modified as the project team understanding of the product increases. Iterations develop the product through a series of repeated cycles, while increments successively add to the functionality of the product [ISO 24765]
Living Documentation	Living documentation is a dynamic method of system documentation that provides information that is current, accurate and easy to understand. Feature files that are written in a natural language format may serve as the core of living documentation. Each file describes how a particular piece of functionality/code is supposed to behave, gives example(s), .30and describes the desired outcome. It can be used for documentation purposes, as part of the acceptance tests and as input for (automated) tests and coding (executable specification). [http://searchsoftwarequality.techtarget.com/definition/living-documentation]
Minimum Viable Product (MVP)	A Minimum Viable Product (MVP) is a product with just enough features to satisfy (early) customers, and to provide feedback for future product development. (See also: Potentially Shippable Increment). [https://en.wikipedia.org/wiki/Minimum_viable_product]
Off-stage actor	Off-stage or indirect actors are further away from the actual functionality to be built. They are not (directly) benefiting from the product, nor do they provide a service. (e.g., governments via laws): the business may not want it; the end-user may not need it. But you should still build it (e.g., privacy concerns). (See also: Actor) [Gojko Adzic, Impact Mapping (2012)]
Pilot project	Project designed to test a preliminary version of a system, process, or method under actual but limited operating conditions and which will then be used to test the definitive version of the system, process, or method. A pilot project should not be trivial, neither should it be critical. [ISO 24765, ISTQB TAE Syllabus]
Policy	Clear and measurable statements of preferred direction and behavior to condition the decisions made [ISO 24765]
Potentially Shippable Increment (PSI)	The increment is the potentially releasable output of the sprint that meets the sprint goal. It is formed from all the completed sprint backlog items, integrated with the work of all previous sprints. The increment must be complete, according to the scrum team's Definition of Done (DoD), fully functioning, and in a

Term / abbreviation	Description and reference
	usable condition regardless of whether the product owner decides to deploy and use it. Whilst originating from the SCRUM methodology this also applies to other ways of workings. Also referred to as Potentially Shippable Product Increment or Potentially Releasable (Product) Increment. (See also: Minimum Viable Product). [https://en.wikipedia.org/wiki/Scrum_(software_development)#Increment]
Primary actor	Stakeholders that are directly benefiting from a product to be built. Usually these are the end users of the product. (See also: Actor) [Gojko Adzic, Impact Mapping (2012)]
Problem space	The goals, actors and impacts or the zone of (what we want to) influence. (See also: Domain) [CARE Syllabus]
Process	A set of interrelated or interacting activities that transforms inputs into outputs [ISO 24765]
Product risk	Risk that a product could be defective in some specific aspect of its function, quality, or structure. [ISO 24765]
Refactoring	Code refactoring is the process of restructuring existing computer code—changing the factoring—without changing its external behaviour. Refactoring improves non-functional attributes of the software. Nowadays the term is also used for Test Automation and other deliverables, sometimes even for deliverables that don't consist of code.
Refining	Refinement is part of the SCRUM process, however principles of it can also be used in non-SCRUM teams/organizations and may be part of the BDD process. It's about getting a User Story ready for a sprint. After refinement the entire team should have a good, shared understanding of the requirements and should give commitment on creating the required functionality. For this the team needs to have a feeling of how complex the story is, usually expressed in a number of 'story points'. "According to the SCRUM guide, refinement takes place during the Sprint Planning-meeting, but in practice many teams have separate Refinement meetings before the sprint planning session".
Requirements	Statement that translates or expresses a need and its associated constraints and conditions. [ISO 24765]
Risk	A factor that could result in future negative consequences. [ISTQB Glossary]

Term / abbreviation	Description and reference
Rule	Constraint on a system specification. [ISO 24765]
Scenario(s)	Step-by-step description of a series of events that occur concurrently or sequentially. Note: A scenario can be a user story, use case, operational concept, or sequence of events the software may encounter. Note: Within Specification by Example the Examples are written in 'Scenario's'. Frameworks like Cucumber or SpecFlow consider these scenario's to be individual, independent automated tests which can be run if implemented using matching step definitions and an underlying test automation framework. A scenario should be independent of other scenario's and should only describe functionality (not implementation). [ISO 24765]
SCRUM	A framework to support teams in complex product development. Scrum consists of Scrum Teams and their associated roles, events, artifacts, and rules, as defined in the Scrum Guide. [https://www.scrum.org/resources/scrum-glossary]
Secondary Actor	Stakeholders needed to make the product work or have an indirect interest. Think about actors like fraud prevention (e.g. legal department), security officer or performance analyst. (See also: Actor) [CARE Syllabus]
Solution Space	Where you are in (the zone of) control by creating the 'Deliverables'. The solution is there to influence the goals, actors, and impacts. (See also Domain) [CARE Syllabus]
SpecFlow	A framework that allows to separate test automation from specifications and provides additional tools to structure your code and consists of Cucumber for .Net with a sophisticated Visual Studio integration. [https://specflow.org/]
Specification by Example	A collaborative approach for defining requirements and business-oriented functional tests for software products based on capturing and illustrating requirements using realistic examples instead of abstract statements. [ISTQB Glossary] Applied in the context of Agile Software Development methods, in particular Behaviour-Driven Development. This approach is particularly successful for managing requirements and functional tests on large-scale projects of significant domain and organizational complexity.

Term / abbreviation	**Description and reference**
Step definition (StepDef)	Step-by-step procedure which will make it possible to convert the simple text 'Steps' from the feature file into code that can be executed. Note: also referred to as 'glue code'. Within BDD tools like Cucumber or SpecFlow, Step Definitions are used to 'glue' the scenarios in Gherkin to the underlying (test) automation. [CARE Syllabus]
Story	1. Simple narrative illustrating the user goals that a software function will satisfy 2. Narrative description of a software requirement, function, feature, or quality attribute, presented as a narrative of desired user interactions with a software system. [ISO 24765] Note: In the context of SCRUM a user story is also a synonym of a piece of work to be developed in a sprint. (See also: User Story)
System	Combination of interacting elements organized to achieve one or more stated purposes. [ISO 24765]
Test Automation	The use of software to perform or support test activities. [ISTQB Glossary] In software testing, test automation is the use of special software (separate from the software being tested: e.g. tools) to control the execution of checks and the comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive tasks in a formalized testing process already in place or perform additional testing that would be difficult to do manually.
Test Automation Framework	A tool that provides an environment for test automation. It usually includes a test harness and test libraries. [ISTQB Glossary] A framework that can be used for one or more levels of test automation in the Test Automation Pyramid. The Framework consists of code (which can be called from the Step Definitions within a BDD tool like Cucumber or SpecFlow) which is used as a driver to call the software under test.
Test (Automation) Pyramid	A graphical model representing the relationship of the amount of testing per level, with more at the bottom than at the top. [ISTQB Glossary] Note: A Test Automation Pyramid describes the number of checks from low level unit tests at the bottom up to high-level end to end (chain) tests at the top. When automating checks, the higher in the pyramid the more time these checks will likely take

Term / abbreviation	**Description and reference**
	to execute and the more brittle they become. Therefore, it is considered good practice to do as much testing as possible as low as possible in the pyramid to get faster and more reliable feedback from (regression) testing.
Test Driven Development (TDD)	A software development technique in which the test cases are developed, and often automated, and then the software is developed incrementally to pass those test cases. [ISTQB Glossary] TDD refers to a style of programming in which three activities are tightly interwoven: coding, testing (in the form of writing unit tests) and design (in the form of refactoring). [https://www.agilealliance.org/glossary/tdd/]
Testing (Software Testing)	The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of a component or system and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. [ISTQB Glossary] Note: The process of validating and verifying that a software program or application or product: • Meets the business and technical requirements that guided its design and development • Works as expected • Can be implemented with the same characteristic. Testing is something that we do with the motivation of finding new information. Testing is a process of exploration, discovery, investigation, and learning.
Three Amigos	A collaboration between three parties: Developer, Tester, and Business (Product Owner). However, currently the term is often used for generally any collaboration/communication where Agile teams and/or other stakeholders are involved. Goal of the 3 (or more!) Amigos is to create a general understanding of what is needed, how this can be created and when it is considered done and of provable sufficient quality. [http://www.sq-mag.com/en/handle/magazin-reader/sq-mag-no-2.html?file=files/magazine/pdf/SQ_mag_No%2002_2.pdf]
TRIMS	Acronym for a set of criteria to be considered while automating tests: "T" Targeted to a specific risk and automated on the lowest layer the testability allows. "R" Reliable: To maximize their value, checks need to avoid flakiness, we need them to be deterministic.

Term / abbreviation	**Description and reference**
	"I" Informative: Passing and failing checks need to provide as much information as possible to aid exploration. "M" Maintainable: Automated checks are subject to constant change, so we need a high level of maintainability. "S" Speedy: Execution and maintenance need to be as fast as the testability allows to achieve rapid feedback loops. [https://automationintesting.com/2019/08/trims-automation-in-testing-strategy.html]
User	Individual or group that interacts with a system or benefits from a system during its utilization. [ISO 24765]
User Story	1. Simple narrative illustrating the user goals that a software function will satisfy 2. Narrative description of a software requirement, function, feature, or quality attribute, presented as a narrative of desired user interactions with a software system. [ISO 24765] Note: In the context of SCRUM a user story is also a synonym of a piece of work to be developed in a sprint. Each user story is expected to yield, once implemented, a contribution to the value of the overall product, irrespective of the order of implementation. [https://www.agilealliance.org/glossary/user-stories/]
Validation	Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. [ISO 24765]
View Model	Outcomes of the event and/or policy are presented to a user or other system: e.g., screen, report etc. (model of the view). [CARE Syllabus]
Verification	Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled. [ISO 24765]
Why-Who-How-What-questions	Basic principle used in Impact Mapping (see Impact Mapping). By asking these questions in this order, the goal, the actors, the impacts, and the deliverables (for development of a system) are successively determined. [CARE Syllabus]

(Intentionally left blank for your notes)