**Glossary of Terms**

**Analysis tools:** The tools offered by ATLAS.ti are the word cruncher, the code cooccurrence tree and table explorer, the query tool and the code-primary documents table. See Chapter 6 for more details.

**Categories and subcategories:** The aim of developing a coding system is to organize the data into main categories and subcategories. Main categories at the end of coding are likely to contain no data. They provide a common label for the subcategories united underneath. Depending on your way of coding, you may at first collect data segments within main category codes. This is likely when using a deductive framework. Then subcategories are built based on the items within this main category container by reviewing them, looking for items that are similar and uniting them under a common subcategory label until all items from the main container have found a place in one of the sub-containers.

When beginning with descriptive-level codes, main category codes are developed via the process of conceptualizing, comparing and contrasting data segments and descriptive code labels, looking for things that are similar and developing new code labels that allow for these segments to be collected under a common name. During this process it is likely that main categories and subcategories are developed at the same time.

Categories may contain more than one level of subcategories, if the main aspect can be subdivided further within the same meaning context. Subcategories should, however, not be built up of different types of aspects like reflecting content, an evaluation or a time. A category that contains three or more levels of subcategories should be closely reviewed to check whether in reality it contains different content layers that may be better coded in different categories.

**Codes:** Keywords that are generally linked to quotations, but do not have to be. You can also create free codes. When you link free codes to a number of different codes, then they are called abstract codes.

**Codes as methodological device:** You turn codes into methodological devices by adding an appropriate label. Here the process of collecting becomes very important. You need to compare and contrast data segments and think of suitable names for data segments that are similar. Thinking of a common name under which similar data segments can be collected is likely to enable the researcher to move away from the descriptive to a conceptual – and over time to a theoretical – level. During this process the researcher begins to develop categories and subcategories.

**Codes as technical device:** Technically speaking, a code is a device that can be attached to a data segment as a label. At the beginning of an analysis, this is a useful first step in gaining an understanding of the data. With progressive analysis, however, codes need to be turned into a methodological device. For this, a human interpreter is needed. Software cannot distinguish between different meanings and levels of codes. It can handle 50 as well as 5000 codes without telling you whether they make sense or not. There is no computer function that can bring order and logic to your collection of codes.

**Code comment:** While coding you can add some notes to this field, and thoughts and questions that occur during the process of first-stage coding. If it becomes clear over time and with further analysis that something should be coded, a code definition should be entered, maybe a coding rule and a sample quote.

**Code definition:** A code definition describes the meaning of a code and how it has been or should be applied to the data. It can contain a coding rule and an example of a typical data segment coded with this code. Writing code definitions helps to improve the methodological rigor of a study. It forces the researcher to think about the meaning of a code in comparison to other codes. It may turn out that the code system contains codes with different labels but more or less the same meaning. These can then be merged under one common label, which is then also easier to define. Writing definitions also helps to develop codes that are clearly distinct from each other so that they can be applied unambiguously.

**Code reference:** This consists of the so-called groundedness and density of a code. The groundedness provides the frequency of how often a code has been applied; the density shows the number of links to other codes.

**Codes-Primary Documents Table:** shows the frequency of codes or code families across documents or document families. It can be exported as an Excel table. It works well as an exploratory tool in combination with the query tool for further in-depth analysis.

**Coding system:** A well-developed coding system describes the data material in all its facets. It shows the main aspects in the data in the form of categories and the variations within a main category in the form of subcategories. The coding system can reflect different types of main aspects depending on the research questions and the aim of the study. These can be the pure content of the data, the layout, the language used, aspects of time, different speakers or actors, evaluations, level of importance, degree of expression, etc.

As a rough guide, computer-assisted coding systems contain on average about 100–250 codes and 12–25 main categories.

**Collecting:** The first codes that are created may just be descriptive labels. These need to be conceptualized with further coding. The aim is to collect similar data segments under a common code label and not to give each data segment a name.

**Conceptualizing:** This refers to the process of (1) moving from descriptive-level codes to conceptual-level codes, and (2) developing subcategory codes based on data segments collected within a too broad abstract code. Conceptual-level codes unite data segments with similar content; they fulfill the criteria of being a properly sized container where all those things are collected that have something in common and that are in some ways different from others.

**Cooccurrence tools:** The cooccurrence tools can be used for a cross-tabulation of codes. Before you run the tool, it is often advisable to set a code family as filter. The quantitative results of the codes co-occurrency table can be exported in the form of an Excel table. The findings often need further exploration in the query tool.

**Document families:** Document families in ATLAS.ti can be thought of as variables. Technically they are a group of documents. You can for instance group all female and male respondents, all teachers, all postmen, all engineers, all moms, all dads, all singles and all married, unmarried or divorced respondents. You can group all documents by a certain month, year, author or source; all documents from company X into a family called Company X; all documents from companies in industry sector X to a family called Sector X; and so on. Families can be created at any time during the analytic process, then modified, renamed or deleted. Their purpose is to serve as a filter. Thus, you can restrict a search to a particular group of documents. This applies to text searches as well as code retrievals. Using the query tool, you can restrict searches by clicking on the scope button and selecting a primary document family as filter. Document families can also be filtered via the main menu option Documents/Filter → Families, and you can access them via the side panel in the Primary Document Manager.

**External document references:** The basic data management concept in ATLAS.ti is that the project file, the HU, does not contain the actual files that you analyze. It only stores an external reference where the source file for each primary document can be found. This potentially allows you to work with large data sets or large data source files like videos.

**Hermeneutic unit (HU**)**:** A data file that stores everything you do to the data, but not the data themselves. An exception is when you work with internal text documents. Then the HU also contains your data. The HU data file has the file extension .hpr7 for version 7 files or .hpr6 for version 6 files. The file type is ‘Hermeneutic Unit’. The file does not have to be stored at a specific location; you can store it wherever you want. As the HU file is just a regular file, you can copy it, move it, delete it or rename it in the file manager, just like any other file.

**Hyperlinks:** These are links between quotations. Quotations can be linked via named relations and thus are first-class relations.

**Internal documents:** When importing transcripts or survey data, you create internal documents. This means these documents are stored within the HU file and are referenced in the Primary Document Manager as: in HU. Further, text files created via the option Documents / New / New Text Document also become internal documents and are also referenced as: in HU.

**In-vivo coding:** For the computer this means that the highlighted characters are used as the code name. In a computer-assisted analysis, in most cases, it does not make a lot of sense to have code words that are the same as the text they code. Generally a bit of context is needed and this requires extending the quotation beyond the characters used as in-vivo code.

**Library:** ATLAS.ti offers two library locations: My Library for single user projects and the Team Library for team projects. The libraries essentially are folders on your computer that are created when you install ATLAS.ti. The default location is a hidden location that cannot immediately be seen in Windows Explorer. The idea is that the library folders are a domain managed solely by ATLAS.ti and the user does not need to be concerned about them. It is, however, possible to move the default library location to a different place using the Library Manager.

**Link:** A link is the line that you draw between two objects in a network view.

**Master HU:** The Master HU is a term I invented for team projects. The project administrator begins by creating the first Master HU. This Master HU is distributed to team members. Team members add their initials to the file name and work on the part that is assigned to them. When they are done with their work, they send their (now) sub-HUs to the project administrator, who merges them into a new Master HU, adds the date, then distributes the new Master HU back to the team members – and so on.

**Memos:** From a purely functional perspective, memos in ATLAS.ti consist of a title, a type and some text. They can be free or linked to other memos, to codes and to quotations. In Chapters 5 and 6, I suggest various way of using memos in ATLAS.ti. Memos are places to write down all sorts of ideas and thoughts. You can use them to remind you of things like what to do next week, what you wanted to ask your supervisor about, what you wanted to discuss with your team members; thus for project planning. And you can use memos as a place to write up your analysis and as building blocks for a later research report. Regard memos in ATLAS.ti as containers for ideas. Do not create a memo for every single idea. If the idea cannot be developed or expanded over time, then consider whether your thoughts might better be entered as a comment for a quotation.

**Method of computer-assisted NCT analysis:** The recurrent components of the methods are noticing things, collecting things and thinking about things. They occur during the process of initial first-stage coding, are repeated when structuring the code list into higher and lower order categories, and play a role again in second-stage coding and also during the further analytic process. Then noticing, collecting and thinking shifts to collecting ideas, interpretations and gaining insights. The NCT process of analysis applies generally when analyzing data with CAQDAS and can be embedded within other methodological approaches. For a detailed description of the method, see Chapter 1.

**Merging:** Merging means combining the contents of various HUs into one HU. You can merge two HUs at a time. The HUs can contain either the same documents or different documents. In the first case, the documents are merged; in the second, they are added. The same applies to codes. The HUs can contain the same codes or different codes. If they contain the same codes, you merge them; if they contain different codes, you add them. If you have some codes that are the same as well as some that are different, you still use the option **Merge**, otherwise all codes that have the same name will be duplicated. This option also allows you to merge only specified objects; for example, you can add only networks or memos and ignore the rest.

**Network views I:** Network views offer a place for visualizing relations within your data. You can link almost all objects to each other; visualize your codings; and visualize relationships between codes, relationships between quotations, relationships between memos and memos, relationships between memos and quotations, relationships between memos and codes, and relationships between object families and their members. Network views can also contain thumbnail images of primary documents (for more details, see Chapter 7).

**Network views II:** Network views visualize the various ATLAS.ti objects and the links between them. Objects that can be included are: primary document codes, quotations, memos, all object families and the network views as icons themselves. Primary documents can be visualized as thumbnail images. The following objects can be linked: codes to codes, codes to quotations, codes to memos, memos to memos and quotations to memos. Families can be visualized as well by showing the links between the family name and each member. The family name cannot be linked to other objects. Links between two codes and between two quotations are considered to be first-class relations as the links can be named. All other links are unnamed and therefore ‘second class’. In NCT analysis, network views play a major role in the conceptual level of analysis, when the analyst begins to see relations in the data during the process of writing research question memos.

**Nodes:** All objects inserted into a network view become nodes. They are visualized as code nodes, memo nodes, quotation nodes, etc., but the generic term applies to all of them. Therefore you need to be careful when deleting a node: you will not just remove the object from the network view, but also delete it from the entire HU.

**Noticing:** Noticing refers to the process of reading or looking through your data (like an explorer walking through an unknown landscape) with the aim of describing the territory in as much detail as possible. The explorer takes out his or her notebook and starts writing down notes or drawing sketches. The qualitative researcher as explorer begins to mark and label segments, creates quotations, adds first codes, writes comments and memos**.**

**Object Managers:** The four main objects in ATLAS.ti are the primary documents, the quotations, the codes and the memos. The list of objects can be viewed in the fly-out panel (version 7), by clicking on the down arrow to open the list field, or you can open a separate window for each object. These separate windows are called ‘Object Manager’. The Object Managers have their own main menu and a toolbar. The options available are repetitions from the main menu, but with one slight difference. When selecting an object in the Object Manager and then one of the menu options, the option only applies to the selected object. Selecting the same option from the main menu has a global effect.

**Operators:** ATLAS.ti offers a total of 14 operators that you can use to formulate a code query:

Boolean: OR, XOR, AND, NOT

Semantic: DOWN, UP, SIBLING

Proximity: WITHIN, ENCLOSES, OVERLAPPED BY, OVERLAPS, FOLLOWS, PROCEEDS, COOCCUR

Within a given project, you will probably never use all operators.

**Primary documents (P-Docs or PDs):** When you add or import a document to your project file, the HU, ATLAS.ti creates a primary document. Each primary document has a name and stores information about the author and when it was created and last modified. In addition, it has some information about where to access the source file. Only when the source is available can the primary document display the content. In version 7 projects, the source files are usually stored in the library. The name of the primary document can be changed within ATLAS.ti without affecting document access. The default name is the name of the source file. Think of the primary document as being like an office assistant who knows all kinds of information about your source files and also where to find them. If the source files are *not* stored at the expected location, the assistant comes back empty handed. This means that you see a blank screen instead of the document content. See also Chapter 3 on project management.

**Primary document comment:** You can write a comment for each primary document. The recommendation is to write meta information about a document into the comment field. For interview transcripts these might be the interview protocols or interview postscripts. Information like age, gender, etc., is managed in primary document families, not the comment field. For other document types, you can use the comment field to specify the source of the document, the context of obtaining the information, a description of who published it, the target audience, and so on.

**Project administrator:** When working in a team, I recommend that one person takes on the role of project administrator. The task of the project administrator is to set up the project, to distribute it to team members, to collect project files from team members and to merge projects.

**Query tool:** The query tool can be used to build queries based on codes or code families and a number of different operators. ATLAS.ti provides three sets of operators: Boolean, semantic and proximity operators. The result of a query formulated in the query tool is a list of quotations. Furthermore, code queries can be combined with variables via the scope button within the query tool. Variables are created via the so-called primary document families (see Chapter 5).

**Quotations:** Marked data segments that have a clearly defined start and end point. Often quotations are coded, but they do not have to be. You can also create so-called free quotations. Free or coded quotations can be used as a source or target to link data segments to each other. Linked quotations are called hyperlinks (see Chapter 7). A quotation has an ID and a name. The ID consists of the primary document number that it belongs to and a number that indicates the sequence of when it was created. The position where a quotation can be found within a primary document is indicated after the name. The quotation name is based on either the first 30 characters of a text quotation or the name of the primary document. This automatically generated name can be modified. If the default number of characters is not sufficient, it can be increased under Tools / Preferences / General Preferences, Tab: General: List Name Size for Quotes.

**Quotation comment:** Technically, the quotation comment field acts as an editor for written text. Potentially, a comment can be added to each quotation. To avoid drowning in too much data, your comments (i.e. short notes) should be written in the comment fields rather than using an ATLAS.ti memo for them. Memos are the place to write down more extensive thoughts and ideas that you elaborate on over an extended period of time (see Chapters 5 and 6).

**Quotation reference:** This is made up of an ID, a name and the position of the quotation in the document. The ID consists of the primary document number that it belongs to and a number that indicates the sequence of when it was created. The quotation name is based on either the first 30 characters of a text quotation or the name of the primary document. This automatically generated name can be modified. If the default number of characters is not sufficient, they can be increased under Tools / Preferences / General Preferences. The location where a quotation can be found within a primary document is indicated after the name. Depending on the document type, different references are used:

**References for audio and video quotations:** hours, minutes, seconds and milliseconds.

**References for Google Earth (GE) quotations:** the geographical coordinates for latitude and longitude (e.g. 51°30′49.21″N, 0°4′41.38″W) as provided by GE.

**Reference for a PDF document:** page number and number of characters on the page for start and end positions. In case the document contains columns, the column number is provided as well.

**References for rtf or doc(x) quotations:** paragraph numbers for start and end positions.

**Relations:** These are the names that you can give to a link. This is possible for code–code links and for quotation–quotation links.

**Relations Editor:** Thisis the window where you can define new relations or modify existing ones. It can be accessed from within a network view via the menu **Links** or via the main **networks** menu.

**Research question memos:** These memos form the building blocks for the results section of a research report. They begin with a well-formulated research question, a description of how the answer to this question was found (e.g. in the form of the query that was run) and how it was developed by the analyst over time, and some linked quotations that provide good examples. These linked quotations might later be used as citations in the research report. Research question memos can be added to the project at any time. Some research questions might be known already from the beginning; others may be added or existing questions may be modified throughout the analytic process.

**Reverse Polish notation:** A code query in the query tool is entered using reverse Polish notation (RPN). This means that you first select the codes or code families and then select one of the operators. Thus, you enter **Code A**, **Code B, OR** instead of **Code A OR Code B**. The logic of reverse Polish notation bypasses the need to learn syntax.

**Role of families in the analytic process:** Primary documents and code families are very handy when it comes to querying the data. You need them as filters to ask focused questions. In addition to the filter options in the side panel of the managers, primary document families can also be set as filters via the scope button in the query tool.

**Super codes:** Super codes represent a saved query. They can be created within the query tool after having clicked on a query. They appear by default in red in the Code Manager and only show frequencies once selected. Each time you activate a super code, the query that it consists of will be run. Thus, a super code is always up to date and changes when the code content that it is based on changes.

**Super families:** When you need a combination of two or more families (e.g. to prepare a special filter for a code query in the query tool or a Codes-Primary Documents Table, or before you run cooccurrence table or tree explorer), you can create super families. This option is available within the Family Managers.

**User account:** All newly created objects in ATLAS.ti are stamped with the user name. The default user is ‘super’. If you want to see your own name instead of ‘super’ in the author field for each object, you need to create a user account (under the Extras menu) and log in using your personal account name. This is a nice but not essential option if working on your own, but a necessity if you work in a team. Based on the name in the author field, each team member can see who has done what.

**Variables:** ATLAS.ti does not offer a spreadsheet where you enter variables like age, gender, profession, location, etc. However, you can use primary document families to represent dichotomous values like male, female, single, married, being a parent or non-parent. By adding a special syntax (::), these can be turned into nominal variables like gender::male and gender::female.