

often existing simulation models are not detailed enough to capture human characteristics.

Several other approaches emphasize the creation of experience: *Experience design*, for instance, is the practice of designing products, processes, services, events, and environments with a focus placed on the quality of the user experience and culturally relevant solutions, with less emphasis placed on increasing and improving functionality of the design [37]. Another example, Dewey's theory of *art as an experience*, says that the entire artistic process should be considered important, not only the physical art-object [38]. Here, the object itself is not the fundamental goal, but rather the development of an "experience" which recaptures some aspect of life. Similarly, the creation of software can be considered an experience, and methods that govern it shape that experience.

III. DEVELOPER EXPERIENCE

This section presents a definition of *developer experience* (DE^x). The definition is influenced by the UX concept. We assume that several factors influence DE^x , which in turn affects outcomes of software development projects. The word "developer" refers here to anyone who is engaged in the activity of developing software, and "experience" refers here to involvement, not to being experienced, although the two are interlinked.

UX has evolved beyond user interface design. Practitioners and researchers know that it is not sufficient to focus only on the user interface and on avoiding usage defects, increasing robustness, and ensuring safety. The user perspective has been shifted first to efficiency and ease of use, then to the question of appropriate use and fitness for purpose, and finally to considering the entire experience of using a (software) product or service. This progression can be thought of as a maturation process in the human-computer interaction field. In Table I, we draw a parallel to the developer perspective, where the end goal is not to use but to create a product or service. Here, the basic level is exemplified by prescriptive process models, the efficiency level by descriptive and adaptive process models, the appropriateness level by a detailed understanding of the process-product relationship in a specific context, and the final level by the entire experience of being a software developer and carrying out software development activities.

As noted, software development is an intellectual activity, which rests on the capabilities of the mind, requiring both thought and motivation to carry out. In psychology, the concept of mind is commonly divided into cognition (attention, memory, producing and understanding language, problem-solving, decision-making), affect (feeling, emotion), and conation (impulse, desire, volition, striving). We explicitly include in DE^x not only affective aspects, but also cognitive, conative, and social aspects of experience. Since the end goal in the developer perspective is to create software, it is

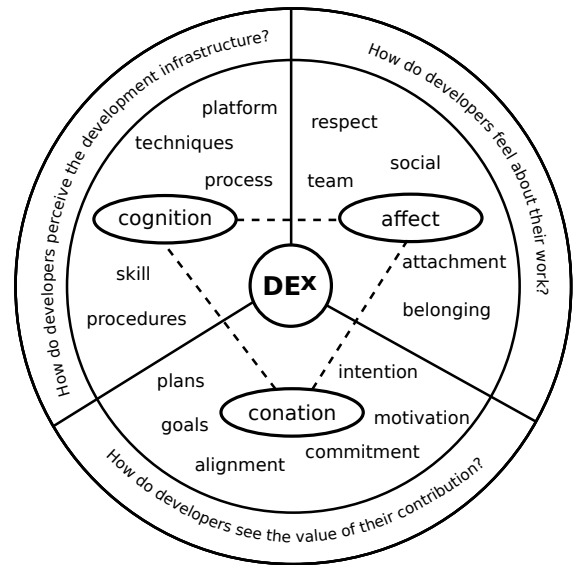


Figure 1. Developer Experience: Conceptual framework.

especially important to consider how thought and feeling is turned into intentional action, and how group work should be systematically organized to support this.

DE^x consists of experiences relating to all kinds of artifacts and activities that a developer may encounter as part of their involvement in software development. These could roughly be divided into experiences regarding i) development infrastructure (e.g. development and management tools, programming languages, libraries, platforms, frameworks, processes, and methods), ii) feelings about work (e.g. respect, attachment, belonging), and iii) the value of one's own contribution (e.g. alignment of one's own goals with those of the project, plans, intentions, and commitment). Figure 1 shows the concept of DE^x as an interaction between cognitive, affective, and conative factors. Each dimension of DE^x consists of a multitude of complex sub-factors. The cognitive dimension consists of factors that affect how the developers perceive their development infrastructure on an intellectual level. This includes concrete interactions with development tools and execution of a software process. Perceiving these in a positive light is likely to contribute to better DE^x . The affective dimension consists of factors that influence how developers feel about their work. Respect and belonging are social factors that work to create a feeling of security. Attachment to persons, teams, or even habits of work also belong to this dimension. Positive feelings in general can be an important factor in good DE^x . The conative dimension consists of factors that affect how developers see the value of their contribution. Intentional, planned activity with personal goals that are properly aligned with the goals of others is likely to increase the sense of purpose, motivation, and commitment, and thus positively affect DE^x .