Chapter 15
Working Towards Fostering Programming Acceptance in the Everyday Lives of Older and Adult People with Low Levels of Formal Education: A Qualitative Case Study

Sergio Sayago, Angel Bergantiños and Paula Forbes

15.1 Introduction

This chapter discusses factors that contribute to the *possible*[1] acceptance (or rejection) of programming in the everyday lives of older and adult people with low levels of formal education. We do this by drawing upon three in-person courses on learning programming. The courses were hands-on introductions to Java, Scratch, App Inventor, and Processing. We conducted these courses with (N = 29) older and adult people with different cultural backgrounds (Spanish, Latin-American, East-European, Asian, and Arabian), over an 8-month period, between 2017 and 2018. We carried out the courses in an adult educational center in a working class neighborhood in Barcelona (Spain).

Technology acceptance is concerned with the factors that help us predict and explain why some technologies are accepted or rejected. With the introduction of digital technologies in multiple facets of contemporary living, and the pivotal role they play in most of them, technology acceptance has become an important and active research area. Technology acceptance research dates back to the 1980s, when

[1] We focus on possible rather than actual acceptance because the results of the case study show that programming and our participants are two worlds apart.

S. Sayago (✉) · A. Bergantiños
Universitat de Barcelona, Barcelona, Spain
e-mail: ssayago81@gmail.com

A. Bergantiños
e-mail: anbeye-94@hotmail.com

P. Forbes
Abertay University, Scotland, UK
e-mail: p.forbes@abertay.ac.uk

S. Sayago (ed.), *Perspectives on Human-Computer Interaction Research with Older People*, Human–Computer Interaction Series,
https://doi.org/10.1007/978-3-030-06076-3_15

the seminal Technology Acceptance Model (TAM) was published (Davis 1986). Since then, TAM-like models, such as TAM2 (Venkatesh and Davis 2000) and UTAUT (Unified Theory of Acceptance and Use of Technology) (Venkatesh et al. 2003) have been published, with the goal of dealing more accurately with technology adoption in a context of growing widespread technology usage. These models have been applied—and validated—in a broad array of workplace/organizational contexts (Marangunić and Granić 2015), accounting for moderate-to-large percentages (between 17 and 70%) of the variance in user intentions to use technologies (Venkatesh et al. 2003). Much of this research is quantitative and has employed self-reported data (Chuttur 2009), which has some limitations—such as not actually focusing on technology usage. In this chapter, we explore technology (in particular, programming) acceptance—to be more precise, possible rather than actual acceptance—in a voluntary (learning) context, which differs considerably from the contexts in which much technology acceptance research has taken place, inasmuch as productivity, effectiveness, and efficiency, which are key constructs in TAM-like models, play a very minor role. We do this by adopting a qualitative approach, based on first-hand observations and conversations, in an attempt to better understand actual technology use and the reasons for the participants' behaviors and intentions.

In recent years, there has been a surge of public interest in promoting computer programming for all. Examples are the European Commission supported initiatives *Code Week* and *All You Need is Code*, along with specialized initiatives intended to introduce programming to school-aged children (K-12) and the launch of introductory programming university courses for students outside Computer Science (Chilana et al. 2016). This has given rise to critical views, which challenge the need of having *everyone* learn to code, e.g. (Shein 2014). Yet, programming, understood either in its traditional, low-level sense (i.e., turning out code) or from the viewpoint of computational thinking, i.e., learning how to think like a programmer, is widely seen as a key skill in the 21st century (Montfort 2016). As stated in (Guo 2017), programming skills can empower older people (65+), who represent a large and fast-growing fraction of the global population, to improve their quality of life, maintain part- or full-time employment, and compensate for the shortage of programming teachers in primary and secondary schools. However, little is known about the relationship between programming and older people. (Guo 2017) is the first known study of older adults learning computer programming. The profile of the older adults who participated in (Guo 2017) was skewed towards highly educated, technology-literate and self-motivated. In this chapter, we are interested in exploring the factors that can foster programming acceptance in the everyday lives of a very different profile of older people than that of those older adults who participated in (Guo 2017). We also enrich the discussion by adding an intergenerational perspective, which is important to understand similarities and differences between older and non-older people.

What factors can help us predict and explain the possible acceptance or rejection of programming among older and adult people with low levels of formal education? We discuss the relative relevance of key technology acceptance constructs, showing that Perceived Ease-Of-Use (PEOU) is much less important than Perceived Usefulness (PU) in fostering programming acceptance. All our participants perceived that they had to discover and understand the fit of programming in their lives, as opposed to those older people who participated in (Guo 2017), in order to decide to explore the technology further. PU has therefore a non-instrumental meaning in our case study. We also show that the figure of the course instructor and the group played a key role in fostering programming acceptance. The social atmosphere turned out to be key to encourage decision-making. Thus, we argue that the predominant focus on the individual in technology acceptance does not seem to predict and explain well enough possible acceptance or rejection in our case study; a shift to social acceptance seems more suitable for doing so. We also discuss some methodological—and ethical—issues, such as the difficulties in asking validated items of TAM (e.g., "I have the knowledge necessary to use the system") to older and adult people with low levels of formal education.

15.2 Overview of Related Works

There is a great deal of research on technology acceptance. In Sect. 15.2.1, we review selected studies, which help us to focus on three key aspects of previous research we aim to highlight in this chapter, because of their connection with the case study: the origins of TAM, its evolution, and criticisms. In Sect. 15.2.2, we turn to computer programming, which is also gaining traction in the HCI community, and discuss the 'for all' aspect from the viewpoint of older people and technology acceptance.

15.2.1 Technology Acceptance

"With growing technology needs in the 1970s, and increasing failures of system adoption in organizations, predicting system use became an area of interest. However, most of the studies carried out failed to produce reliable measures that could explain system acceptance of rejection" (Chuttur 2009, p. 159). To fill this gap, Fred Davis, in 1985, published the Technology Acceptance Model (TAM), wherein the user's attitude toward a system use is influenced by two major beliefs: Perceived Usefulness (PU) and Perceived Ease Of Use (PEOU). PU and PEOU are, respectively, originally defined as "The degree to which a person believes that using a particular system would enhance his or her job performance" and "The degree to which a person believes that using a system would be free of effort". The origins of TAM can be traced to a psychological theory, the Theory of Reasoned Action (TRA 2018). According to this theory, intention to perform a certain behavior precedes the actual behavior, and behavioral intention is a function of both attitudes and subjective norms, which are defined as a person's perception that most people who are important to him or her think s/he should or should not perform the behavior in question.

As stated in (Marangunić and Granić 2015), consistent findings that PU was a major determinant of the intention to use gave rise to an extended model, named TAM 2 (Venkatesh and Davis 2000), which sought to identify the variables that influence PU. The variables included are subjective norm, image (the desire of the user to maintain a favorable standing among others), job relevance (the degree to which the technology was applicable), output quality (the extent to which the technology adequately performed the required tasks), and result demonstrability (the production of tangible results). The Unified Theory of Acceptance and Use of Technology (UTAUT) (Venkatesh et al. 2003) is another important extension of TAM, formulating a unified model that integrates elements across eight models, including TRA and the Diffusion of Innovation of Rogers (2003). More recently, and prompted by the realization that new contexts of technology use might result in changes in theories, an extended version of UTAUT, named UTAUT 2 (Venkatesh 2012), has been proposed to study acceptance and use of technology in a consumer context, where price and hedonic motivation, such as fun and entertainment, are important factors. As stated in the introduction, several reviews indicate that TAM-like models have been validated in different workplace/organizational contexts (Marangunić and Granić 2015; Chuttur 2009). Yet, and despite a growing ageing population, older people have been mostly overlooked (Chen and Chan 2011; Comunello et al. 2015). In an attempt to fill this gap, and in light of the predictive and explanatory power of TAM-like models, an extended TAM for older people in gameplay contexts has been proposed (Wang and Sun 2016). The factors influencing acceptance of technology for aging in place have also been examined—although very little of this research has considered TAM-like models (Peek et al. 2014).

TAM has also been criticized, despite its predominant role in technology acceptance research. According to (Benbasat and Barki 2007), TAM-based research has paid scant attention to the antecedents of its belief constructs, failing to understand what actually makes a system useful, and the efforts to "patch-up" TAM in evolving IT contexts have not been based on solid and commonly accepted foundations, resulting in a state of theoretical confusion and chaos. In Chuttur (2009), it is stated that one of the main criticisms is that self-reported use data are used to measure system use instead of *actual* use data. Another important limitation, suggested and discussed in Chuttur (2009), is that TAM is a deterministic model, overlooking the fact that evaluation and reflection might direct a person to reformulate his or her intention, or even to take a different course of action. This is echoed in Benbasat and Barki (2007) by focusing on the "dynamic interplay" of the various behaviors that revolve around system use.

In this chapter, we do not aim to replicate, validate or propose a new TAM model. This is beyond the scope of this chapter. Our objective is to identify and discuss key constructs of TAM and TAM-like models—and to suggest new ones, if any—that aid in predicting and explaining the possible acceptance (or rejection) of programming among a potential, and mostly overlooked, group of programmers.

15.2.2 Computer Programming

Computer programming is no longer the domain of *programmers*. As stated in the Introduction, there is an increasing interest in opening up programming to everybody. However, computer programming for all falls behind its inclusive goal. Older people have been overlooked in most of the efforts, which tend to focus on kids and young people, and have been conducted at different levels, ranging from public initiatives to technological developments, such as Scratch[2] and App Inventor,[3] and research studies (Kafai and Burke 2014). A noteworthy exception is Guo (2017), which is based on an online survey on the motivations, learning practices, and frustrations of approximately 500 English-speaking older people (over 90% were Managers, Professionals or Technicians), from 52 different countries, who were learning programming by using an educational website. Guo (2017) shows that making up for missed learning opportunities during youth, keeping their brains challenged, and implementing a specific hobby project idea were the respondents' top three motivations for learning programming. The respondents also reported using free online resources, mostly MOOCs, blogs, and web tutorials. The three most important reported learning frustrations were bad pedagogy, cognitive impairments and no human contact with tutors or peers. Guo (2017) does not discuss the results in terms of, or by taking into account, technology acceptance constructs, despite its connection. We make this connection explicit, and discuss it, in this chapter.

15.3 The Case Study

In Sect. 15.3.1 we situate the case study in its context. In Sect. 15.3.2, we set out its key objective and provide an overview of how we conducted the case study and the profile of the participants who took part in it. In Sect. 15.3.3, we summarize the courses and present key aspects of the setting where we conducted them. In Sect. 15.3.4 we deal with data gathering and analysis.

15.3.1 Context

We carried out this case study within the context of the AGORA 4.0 project (https://agora4.org/), which aims to democratize technologies related to digital making amongst older and adult people in risk of social exclusion. In Barcelona, there is a growing interest in fostering digital making amongst its citizens. Examples are FabLab Barcelona,[4] which belongs to the fab.city global initiative,[5] intended to make locally productive, globally connected cities and citizens, and the *Ateneus de Fabricació* (in English, Athenaeum of Fabrication), which are a public service designed to disseminate digital making in society. These examples share the democratization goal

[2] https://scratch.mit.edu/. Accessed November 10, 2018.

[3] http://appinventor.mit.edu/explore/. Accessed November 10, 2018.

[4] https://fablabbcn.org/. Accessed November 10, 2018.

[5] https://fab.city/. Accessed November 10, 2018.

of the international digital making movement (Ames et al. 2014; Dougherty 2012). Making is also seen as enabling marginalized groups to participate in computing innovation (Lindtner et al. 2016). Yet, prior works point out that making in practice often falls short of its ideals, because those who participate in maker communities are mostly from the middle and upper classes, and the presence of women and minority populations remains low (Ames et al. 2014; Meissner et al. 2017). A noteworthy exception is Meissner et al. (2017), which reports on a qualitative study of making with people with disabilities. In AGORA 4.0, we have realized—in informal conversations and visits to the *Ateneus de Fabricació*—that a similar situation happens, as people at risk of social exclusion do not tend to participate in digital making activities, due to a number of factors, ranging from a lack of interest or technical knowledge to feeling 'like a fish out of water' and economic reasons.

15.3.2 Objective, Implementation and Profile of the Participants

The case study aims to contribute to this goal of digital democratization by focusing on programming, which is important for enabling participation in some digital making practices, such as modifying or building digital artifacts.[6] Working towards this end, we have carried out 3 free courses intended to provide older and adult people with low levels of formal education with a hands-on introduction to programming. The participants included a mix of older and adult active computer users. Most of the participants (23) were original from Spain. A few of them (6) were from Latin America, Eastern Europe, Asia and Arabic countries.

- Total number of participants: 29
- Age ranges: <30: 4; 31–40: 2; 41–50: 2; 51–60: 7; 61–70: 12; 71–80: 2
- Sex: 18 men, 11 women
- Participants/course: 6 participants in the first course; 11 in the second; 12 in the third. No participant took more than one course.

The courses focused on Java,[7] Scratch, App Inventor, and Processing.[8] Although the choice of the programming language or environment is not as important in this context as programming per se, we choose JAVA because of its popularity (TIOBE 2018), its connection with smartphones—via Android—and our previous experience of teaching it to undergraduate students. Scratch and App Inventor were chosen because of their apparent ease-of-use, they are block-based programming tools, and their connection, especially App Inventor, with mobile apps and smartphones, which are very popular nowadays. We also explored Processing because of it creativity

[6]Programming is also important in End-User Development (Díaz et al. 2015), which, in our view, is related to making in the sense of ordinary people creating their own technologies.

[7]https://www.java.com.

[8]https://processing.org/. Accessed November 10, 2018.

and visual aspect, being a programming language targeted at designers, with little experience of programming.

15.3.3 Courses and Setting

The in-person courses, which lasted between one and three months, were conducted by the first and second author. Overall, the structure of the courses was similar, with slightly different implementations, depending on the course and the participants' interests. We ran practical sessions, in which we conducted live coding and asked the participants to make predictions (i.e., what they thought the program would do). The participants also did a number of classical exercises (e.g., writing a program to check if a number is odd or even) by programming individually and in pairs, and solving Parsons Problems (where chunks of code have to be placed in the correct order). We followed a number of tips for teaching programming "at any level and to any audience" (Brown and Wilson 2018, p. 1).

We conducted these courses in Àgora (AG), an adult educational center in Barcelona, Spain, and partner of the project (see Fig. 15.1). These courses had the same format (weekly sessions of 2-h long) as the other courses on computers in AG, which has been operating for almost 40 years. Since the 1980s, AG has been fostering the social and digital inclusion of people who are, or might be, excluded from the Catalan society, such as immigrants and older people. To this end, AG adopts a dialogical learning approach, which empowers the students—using AG terminology, participants—to decide what they want to learn in free courses. AGORA conducted the recruitment of the participants, and written consent forms were obtained.

Fig. 15.1 Participants in one of the programming courses

15.3.4 Data Gathering and Analysis

We took paper and computer notes of our conversations and observations immediately after the sessions, which were so active that they hindered in situ note-taking. We then analysed these notes by reading them every week and finding key topics related to technology acceptance constructs. We wrote working drafts of the results (and this chapter) and shared them with the authors of this chapter and colleagues who did not take part in the fieldwork. These drafts were updated until the authors agreed on the results.

15.4 Relevant Findings

PU and PEOU are two key constructs in TAM-like models. In Sect. 15.4.1, we discuss their relative relevance in the case study. While technology acceptance has a strong focus on the individual, in Sect. 15.4.2 we highlight the importance of two key social aspects in the participants' possible acceptance of computer programming: the group and the course instructor. In Sect. 15.4.3 we discuss an important implementation aspect—the order in which programming tools are introduced in courses—and suggest an important construct in technology acceptance in this context: perception of exclusion. In Sect. 15.4.4, we present some difficulties in conducting questionnaires to measure possible technology acceptance constructs with older people with low levels of formal education.

15.4.1 PU is More Important than PEOU

Overall, observations and conversations confirmed that PU is far more important than PEOU in fostering programming acceptance. This relative relevance is exemplified in three key aspects.

Firstly, when asked about the reasons for taking on the courses, participants answered that they were there "*to know what programming is*", "*to know what I can do with programming*", and "*to learn more about computers and the Internet*". Our interpretation of these answers is that they were finding out the usefulness—in a broad sense, 'what it is and what I can do with it'—of programming.

Secondly, although learning programming was not easy, participants' attendance was very regular, and they felt unease when they could not come. During the courses, we observed that learning programming was far from straightforward for them. Their most common difficulties were to write programs without syntactical errors—common errors were missing symbols, such as a semicolon, and parenthesis or brackets—and turning a problem into a representation the computer understands (i.e., a program) by using key structures, such as conditional (if-else) and iterative (while, for) statements. Yet, these difficulties did not put our participants off taking on the courses.

Thirdly, as the participants were gaining more programming experience, their questions and comments highlighted the usefulness—namely, the *fit*—of programming in their lives. For instance, some participants showed us exercises, which related to their interests, they had done at home in order to know more about programming and what they could do with it. Others told us that they were trying to see the role or usefulness of programming in their everyday lives. In this sense, participants, especially the younger ones, and those with more experience with computers, wanted to know things that mattered to them, for instance, how to program a game or a chatbot in their smartphone.

The relative relevance of PU over PEOU was also visible in programming rejection. Throughout the project, we met participants who took the first course. These participants did not enrol on any other courses on programming. Our conversations with them stressed the importance of PU. As one of them told us "*the course was interesting, and you're a great teacher, but I don't see the fit of programming in my life. It's something far removed from my life—at least now*".

15.4.2 Social Technology Acceptance: The Group and the Course Instructor

Two social aspects stood out in working towards programming acceptance among our participants: the group and the course instructor. The group played a key role in sustaining participants' interest over time. Learning computer programming in the company of others, sharing their comments and doubts, supporting and learning from each other, was very important for the participants to learn and discover programming, which is a key step towards acceptance. As acknowledged by them, "*Being here is very important. I don't see myself learning programming alone at home*". Within this context, another significant social aspect was the figure of the course instructor.[9] In the courses, the instructors were the first two authors of the chapter. They were the people responsible for creating the learning materials, which are available online (agora4.org), and running the sessions, and were faced with a number of important difficulties. What is an authentic learning activity in programming in this context and with this profile of participants? How does one explain what a variable, a program, a conditional statement, debugging… is to older and adult people with low level of formal education? They could not fall back to their—mostly, the first author's—previous experience of teaching other computer-based technologies to older people, as they did not know either the interests or needs of our participants. Our technological background did not help us either, because we found it very difficult to find appropriate examples and exercises, understandable terminology…Most of these difficulties

[9]This finding is more related to the authors' own reflection on their impact on the case study than on participants' comments and views.

were not overcome at the end of the courses. Yet, by being patient and attentive to participants' interests and needs, and encouraging them to participate in the sessions (by voicing their views, opinions…and definitions!), the instructors managed to run the sessions smoothly and maintain participants' motivation throughout them.

15.4.3 POE (Perception of Exclusion)

The possible acceptance of programming, and some programming tools, was either hindered or fostered by the order in which they were introduced in the courses. We were not aware that the order mattered. However, the order of introduction helped us identify a new (or different) construct in technology acceptance—perception of exclusion, which we define as the degree to which a person believes that using a technology makes them become or feel like an 'extraordinary' computer user. In the first course, participants programmed in JAVA with Netbeans, a professional IDE, and Scratch was introduced at the end of the course. Participants' refused to use Scratch, on the grounds that it was perceived as something too simple that made them feel stupid. In the other courses, however, we introduced Scratch—and another block-based programming tool, App Inventor—during the first weeks. We did so because we considered that both could be a smooth, visual and useful introduction to programming. The participants' acceptance of both was positive, confirming our hypothesis. Some of the reasons stated were simplicity and clarity. Their opinion did not make them refuse other, more professionally looking programming tools, such as the Processing IDE.[10]

15.4.4 'I Consider I Have the Knowledge Necessary'

In a previous project, which was about digital games and older people, we explored the acceptance of digital games. We did so by using questionnaires, which consisted of Likert scales of validated technology acceptance constructs, such as "I have the knowledge necessary to use the system". In addition to modifying these constructs, which make reference to productivity and efficiency issues, we found, as one could expect, that our participants were 'insulted' when asked about their knowledge. Prompted by the heterogeneity of the older population, we wanted to know if a similar behavior could be exhibited by a different group of participants in a different context. In the first course, we also attempted to administer a questionnaire based on TAM constructs, with very similar comments. For instance, "*I think some people can take offense if you ask them about their knowledge. We know we don't have much knowledge of many things (…)*" We did not ask the participants to complete any questionnaires about acceptance in the other courses. Instead, we talked to them, which was far more natural, and easy for them.

[10]http://download.processing.org/processing-3.3.7-windows32.zip. Accessed November 10, 2018.

15.5 Discussion

This chapter has explored technology acceptance in a, perhaps, rather unusual way. To begin with, given that the case study has been conducted in a learning scenario, it could be argued that this chapter has not explored technology acceptance but learning motivations and difficulties. While we acknowledge the likely and unavoidable overlap,[11] the results show that a voluntary learning scenario is very rich in terms of technology acceptance, as it helps us identify potential factors that contribute to foster programming acceptance amongst older and adult people with low levels of formal education. In light of the results presented, there are also reasons to argue that *actual* programming acceptance is very difficult to explore in other contexts with this profile of participants, as programming is very far removed from their everyday lives, and they first need to discover it. In addition to this, while technology acceptance research traditionally focuses on a single system, which system have we explored? We have not focused on any programming tool in particular, as the challenge was to start to understand the almost overlooked relationship between our participants and programming. Having focused on a single programming tool could have provided, in our opinion, a very partial and limited view.

In the first three subsections (Sects. 15.5.1, 15.5.2 and 15.5.3), we discuss the three key contributions this chapter makes to HCI research with older people. In Sect. 15.5.4, we discuss some implications that can be drawn from the results, and in Sect. 15.5.5, important limitations.

15.5.1 An Intergenerational Case Study of Fostering Programming Acceptance

As discussed in Sect. 15.2, technology acceptance research has mostly overlooked older people. To the best of our knowledge, this chapter is the first study that explores factors that can contribute to foster programming acceptance amongst older and adult people with low levels of formal education. Despite the evident heterogeneity of older people, which is often alluded to in older-adult HCI, the results show that our participants, with different cultural backgrounds and age ranges, do not differ considerably as far as possible programming acceptance (or rejection) is concerned. This chapter has adopted a qualitative approach, which does not predominate—an exception is Peek et al. (2014)—in technology acceptance research, which is heavily dominated by questionnaires and surveys. The results show that a quantitative approach can prove to be very difficult—if not impossible—with the profile of participants. This qualitative approach has been implemented in a learning and voluntary scenario, which differs considerably from the mandatory, workplace/organizational contexts where technology acceptance research tends

[11] Learning is an item of PEOU, e.g. “Learning to operate the system would be easy for me” (Venkatesh et al. 2003).

to be conducted. This scenario also differs from previous studies of technology acceptance for ageing in place (Peek et al. 2014). This learning scenario, which has prompted us to focus on possible rather than actual acceptance, corresponds to an "in the wild" situation (Rogers and Marshall 2017), which is gaining traction in HCI, and is in accordance with changes in TAM (e.g. UTAUT2), which have been prompted by changes in contexts of technology usage.

15.5.2 Relative Relevance of Technology Acceptance Constructs

We have revealed and explained that PU is far more important than PEOU in fostering programming acceptance (and rejection) among our participants. In particular, we have argued for changing "usefulness" for *fit in their lives*, which is a more suitable expression or concept in the context of our case study, as usefulness seems to be narrowed down to 'getting things done' in technology acceptance research. The concept of 'fit in their lives' shows the importance of a purposeful interaction with the world and activity, which are key elements of a theoretical foundation in HCI—Activity Theory (Kaptelinin and Nardi 2006).

We have not seen other technology acceptance constructs, such as price, computer anxiety, society norms, and gender, playing an important role in our results. Nevertheless, this is not to say that they play no role—technology acceptance research dates back from the 1980s and it would be risky to make such a claim. This lack of importance might be due to the yet-to-be explored relationship between programming and older and adult people with low levels of formal education. Future research can deepen and widen our results.

15.5.3 New (or Different) Elements of and Constructs in Technology Acceptance

We have highlighted the importance of the group and the course instructor. The former is, or can be seen as being, loosely tied to the subjective norm element of TAM-like models, in the sense of 'the impact of others on my acceptance'. The latter is, to the best of our knowledge, not acknowledged in any TAM-like models. This reinforces previous claims, which argue for looking at social acceptance (Benbasat and Barki 2007), and one of the key elements in the diffusion of innovation: a social system (Rogers 2003). The importance of the group and the course instructor are two concrete and practical examples of this social side of technology acceptance beyond subjective norms.

We have also stressed the need to consider Perceived Exclusion (PE), which has not been addressed in any TAM-like model we are aware of. On the one hand, the addition of this new construct reinforces claims about the lack of a consolidated theory in TAM (Benbasat and Barki 2007). On the other hand, this construct can be taken as an opportunity to keep improving our understanding of technology acceptance and, eventually, to formulate—if it can exist at all—a general, or more inclusive, theory of technology acceptance. Working towards this goal, it is interesting to note that our PE is similar to the stigmatization found in a systematic review of technology acceptance for ageing in place (Peek et al. 2014).

15.5.4 Some Implications

The results show a number of concrete aspects, such as the importance of the fit of programming in people's lives, the need to consider both individual and social acceptance, and the relevance of the order in which some programming tools are introduced, that can both help us to foster programming acceptance among people who are not usually regarded as programmers and contribute to either achieve or reinforce the inclusive aspect of computer programming for all initiatives.

The results show that computer programming for all can be approached not only from the perspective of learning but also from the viewpoint of technology acceptance—understanding technology not as a single system but in a broader, and, perhaps, richer, sense. The results also show that doing so helps us understand further the—arguably complex—relationship between older and adult people with low levels of formal education and computer programming. Our results differ considerably from those discussed in Guo (2017). Yet, taken together, the results provide a richer picture of computer programming and older people.

The current or predominant perspective on HCI research with older people discussed in the book does not seem to account well enough for the type of technology acceptance discussed in this chapter, as the focus on help and compensation does not reflect well enough other aspects—such as PE and fit in their lives—that come into play into our participants' possible acceptance of programming. The current—third—wave of HCI research seems more suitable for doing so.

Older people might not be such a heterogeneous user group as they might be in other contexts as far as programming acceptance is concerned. As discussed in Sect. 5.1, we have not identified major differences amongst our participants. This apparent homogeneity suggests that the issues discussed in the chapter can apply to several potential adopters of programming, regardless of their chronological age—and even cultural background.

15.5.5 Main Limitations

We have focused on possible and not actual programming acceptance. While this can be seen as an important limitation, our results suggest that we are still far from being able to explore actual programming acceptance among older and adult people with low levels of formal education.

The data gathered does not allow us to claim that our participants will adopt programming after the courses. Yet, these courses have provided them with a hands-on introduction to an unknown technology for them, and this might pave the way for further exploration, acceptance and use. This is not a small result; the findings of the chapter show that our participants and programming are thus far, two worlds apart.

We do not claim that our results can be generalized to other contexts, either online or f2f, and users. This was not our objective, as we aimed to understand technology acceptance within a particular case. Methodologically speaking, a case study is not the best method to argue for general results. Yet, this is not to say that the results cannot apply to other contexts. For instance, we ran a number of 3D printing courses within the context of AGORA 4.0 with different users, and we found very similar results. With the exception of the first author, none of the others was involved in the 3D courses. Further—perhaps, a combination of qualitative and quantitative—research can validate our findings.

Another relevant limitation is the interplay of constructs in programming acceptance (or rejection). We have not addressed the extent to which a construct modifies or determines another. Nor have we examined the temporal aspect of programming acceptance or the consequences of doing so. Adopting—or, in the participants' words—finding the fit of a completely new technology into one's lives takes time, and this process is, arguably, iterative, dynamic, and not deterministic. Perhaps, the complexity of the relationship between programming and older and adult people with low levels of formal education makes it difficult to explore these aspects in a single case study.

15.6 Conclusion

This chapter makes four distinguishing contributions to HCI research with older people. This chapter

- discusses factors that can contribute to help us predict and explain, and foster, programming acceptance among older and adult people with low levels of formal education,
- discusses the relative relevance of important technology acceptance constructs in a relatively unexplored context, and with a mostly overlooked profile of people,
- suggests new elements and constructs to better understand and encourage programming acceptance among older and adult people with low levels of formal education and different cultural backgrounds,
- discusses some methodological issues, which reinforce the need to adapt research methods to older people discussed in chapters of this book.

These contributions introduce a new perspective not only on HCI research with older people, especially that concerned with programming, but also technology acceptance research.

In terms of future perspectives, we plan to look at the results of the courses on 3D printing, and other activities carried out in the AGORA 4.0 project, such as public events in the local neighborhood, from the perspective of technology acceptance, and write the final report of the project. We also aim to understand further the model of Diffusion of Innovations, within which programming can be understood as an *innovation*, which is *communicated* through certain channels over *time* among the members of a *social system*. Addressing these issues seems to provide a complementary and interesting explanation for some of the results of our case study, and operationalized ways of democratizing further programming.

Acknowledgements We are indebted to l'Escola d'Adults de la Verneda—St. Martí, l'Associació de Participants Àgora, and to all our lovely participants, for their interest in the project and allowing us to collaborate with them, share their views and opinions in publications related to our research, and for helping us to keep learning more about the 'older' side of digital technologies. We also acknowledge the outstanding tasks carried out by Pau Blanco, Irene Sainz, Rosa Lloret and Mª Jesús Quesada in the 3D courses. We thank Josep Blat and Mireia Ribera for their comments on earlier versions of the chapter. We acknowledge the support from the Barcelona city council.

References

All you need is code (2018). http://www.allyouneediscode.eu/. Accessed 10 Nov 2018
Ames M, Bardzell J, Bardzell S et al (2014) Making cultures: empowerment, participation, and democracy—or not? In: CHI 2014 extended abstracts on human factors in computing systems (CHI EA 2014), pp 1087–1092
Benbasat I, Barki H (2007) Quo vadis, TAM? J Assoc Inf Syst 8(4):211–218
Brown N, Wilson G (2018) Ten quick tips for teaching programming. PLoS Comput Biol 14(4):1–8
Chen K, Chan, AHS (2011) A review of technology acceptance by older adults. Gerontechnology 10(1):1–12
Chilana P, Singh R, Guo P (2016) Understanding conversational programmers: a perspective from the software industry. In: Proceedings of the 2016 CHI conference on human factors in computing systems (CHI 2016), pp 1462–1472
Chuttur M (2009) Overview of the technology acceptance model: origins, developments and future directions. Sprouts: Working Papers on Information Systems, vol 9
Code Week (2018). http://codeweek.eu/. Accessed 20 Nov 2018
Comunello F, Mulargia S, Belotti F et al (2015) Acceptance of ICTs by older adults: a review of recent studies. In: ITAP (international conference on human aspects of IT for the aged population), pp 239–249
Davis FD (1986) A technology acceptance model for empirically testing new end-user information systems: theory and results. MIT Sloan School of Management, Cambridge, MA
Díaz P, Pipek V, Ardito C et al (2015) End-user development. Springer, Cham
Dougherty D (2012) The maker movement. Innovations 7(3):11–14
Guo P (2017) Older adults learning computer programming: motivations, frustrations, and design opportunities. In: Proceedings of the 2017 CHI conference on human factors in computing systems (CHI 2017), pp 7070–7083
Kafai Y, Burke Q (2014) Connected code: why children need to learn programming. The MIT Press, Cambridge, MA
Kaptelinin V, Nardi B (2006) Acting with technology: activity theory and interaction design. The MIT Press, Cambridge, Massachussets
Lindtner S, Bardzell S, Bardzell J (2016) Reconstituting the utopian vision of making. In: Proceedings of the 2016 CHI conference on human factors in computing systems (CHI 2016), pp 1390–1402

Marangunić N, Granić A (2015) Technology acceptance model: a literature review from 1986 to 2013. Univ Access Inf Soc 14(1):81–95
Meissner J, Vines J, McLaughlin J et al (2017) Do-it-yourself empowerment as experienced by novice makers with disabilities. In: Proceedings of the 2017 conference on designing interactive systems (DIS 2017), pp 1053–1065
Montfort N (2016) Exploratory programming for the arts and humanities. MIT Press, Cambridge, MA
Peek S, Wouters E, van Hoff J et al (2014) Factors influencing acceptance of technology for aging in place: a systematic review. International Journal of Medical Informatics 83(4):235–248
Rogers E (2003) Diffusion of innovations. Free Press, London
Rogers Y, Marshall P (2017) Research in the wild. Morgan & Claypool, USA
Shein E (2014) Should everybody learn to code? Communications of the ACM 57(2):16–18
TIOBE index (2018). https://www.tiobe.com/tiobe-index/. Accessed 10 Nov 2018
TRA (2018). https://en.wikipedia.org/wiki/Theory_of_reasoned_action. Accessed 10 Nov 2018
Venkatesh V (2012) Consumer acceptance and use of information technology: extending the unified theory. MIS Q 36(1):157–178
Venkatesh V, Davis FD (2000) A theoretical extension of the technology acceptance model: four longitudinal field studies. Manag Sci 46(2):186–204
Venkatesh V, Morris MG, Davis GB et al (2003) User acceptance of information technology: toward a unified view. MIS Q 27(3):425–478
Wang Q, Sun X (2016) Investigating gameplay intention of the elderly using an extended technology acceptance model (ETAM). Technol Forecast Soc Chang 107:59–68

Sergio Sayago (Ph.D., Human-Computer Interaction/Computer Science, Universitat Pompeu Fabra, Barcelona) has been conducting HCI research with older people since 2004. He has conducted long-term ethnographic, participant observation and co-design studies of real-life web accessibility (WCAG) and technology use, mostly communication and digital games, in different cultural settings. His current research focuses on long-term empowerment and democratization, addressing making, programming, and (embodied) conversational interfaces, with older and adult people with low levels of formal education.

Angel Bergantiños (BSc, Computer Science with a minor in Mathematics, Universitat de Barcelona) is a computer engineer. He has worked on a number of small projects on machine learning and gamification. He participated in the AGORA 4.0 project (2017–2018), funded by the Barcelona City Council. He developed the accessible website of the project and ran courses on programming for older and adult people with low levels of formal education.

Paula Forbes is a post-doc researcher in the School of Design and Informatics at Abertay University (Scotland). Her main interests are user research and user experience. Paula has participated in a number of interdisciplinary projects across the field of HCI, older people, sustainability and e-health. Paula has several years of experience of working with older people to support their technology learning, after setting up the Dundee User Centre.