



Review Article

Volume 4 Issue 2 - June 2017
DOI: 10.19080/OAJNN.2017.04.555632

Open Access J Neurol Neurosurg

Copyright © All rights are reserved by Pieter L Kubben

Teaching Computer Programming to Medical Doctors, Nurses and Hospital Staff: A Pilot Study



Pieter L Kubben^{1,2*}, Pepijn Looije³, Albert JJA Scherpbier⁴ and Frits van Merode⁵

¹Department of Neurosurgery, Maastricht University Medical Center, The Netherlands

²Department of Medical Information Technology, Maastricht University Medical Center, The Netherlands

³University of Amsterdam, The Netherlands

⁴Faculty of Health, Medicine and Life Sciences, Maastricht University Medical Center, The Netherlands

⁵CAPHRI School for Public Health and Primary Care, Maastricht University Medical Center, The Netherlands

Submission: March 07, 2017; **Published:** June 05, 2017

*Corresponding author: Pieter Kubben, Department of Neurosurgery, Maastricht University Medical Center PO Box 5800, 6202 AZ Maastricht, The Netherlands, Email: p.kubben@mumc.nl

Abstract

There is plenty of opportunity where the medical world can benefit from caregivers who can translate point-of-care challenges into digital solutions using coding skills. The objective of this pilot study is to evaluate the feasibility of teaching computer programming to hospital staff with no or limited programming experience. In October 2015 during the EU Code Week 2015 we organized a 2-day introductory workshop Building Medical Apps for hospital staff (n=14) using Apple's XCode integrated development environment with the Swift programming language. The first day focused on basic programming fundamentals, whereas the second day focused on building a graphical user interface and adding interactivity with code. General programming knowledge improved with a mean of 1.4 ± 0.5 and a median of 1.0 (IQR 1.0) points. App building skills improved with a median of 0.9 ± 0.5 and a median of 1.0 (IQR 0.0) points. The degree to which participants considered having programming skills to be important increased with a mean of 1.2 ± 0.9 and a median of 1.0 (IQR 2.0) points. The willingness to develop apps if participants were to be supported in legal aspects of software development increased with 1.0 ± 0.7 and a median of 1.0 (IQR 0.0) points. In conclusion, this pilot demonstrates feasibility of teaching computer programming to an audience of medical doctors, nurses and hospital staff. Two days of training was sufficient to get a basic level of skills to build a simple mobile app. Participants expressed a desire to continue using these skills, provided they would get support from the hospital such as a helpdesk for questions and regulatory affairs.

Keywords: Programming; Coding; Apps; Mobile

Objective

"Today we live in a world that has been affected by rapid advances in technology. The way we work, communicate, shop and think has changed dramatically. In order to cope with these rapid changes and to make sense of the world around us, we need to not only develop our understanding of how technology works, but also develop skills and capabilities, that will help us to adapt to living in this new era. Learning to code helps us to make sense of how things work, explore ideas and make things, for both work and play. What's more, it helps us to unleash our creativity and work collaboratively with wonderful people both near us and all over the world [1]."

As part of the Digital Agenda by the European Commission, the EU Code Week has been launched in 2013 to promote teaching computer programming to a wide audience. The second

edition involved more than 4,000 coding events in 37 countries, in which more than 150,000 people participated. Similar events are organized during the Computer Science Education Week, in particular with The Hour Of Code initiative [2,3]. There is plenty of opportunity where the medical world can benefit from caregivers who can translate point-of-care challenges into digital solutions using coding skills. The objective of this pilot study, conducted during the EU Code Week 2015, is to evaluate the feasibility of teaching computer programming to hospital staff with no or limited programming experience [4].

Methods

In October 2015, we organized a 2-day introductory workshop Building Medical Apps for hospital staff. Participation was free of charge, expenses were covered by the hospital.

The workshops were organized by the first author, who is a neurosurgeon and an experienced app developer. We used Apple's XCode 7 with the Swift 2 programming language (Apple Inc., Mountain View, CA) [5,6]. 14 people participated in this pilot study and all participants completed the 2-day training. Every participant had their own computer available, which was provided by the hospital during the course.

On the first day, XCode's "playgrounds" were used to introduce fundamental programming concepts (variables, calculations, conditional statements and repetitive statements) without the need for compiling or a graphical user interface. Course material was made available online and participants could complete the exercises at their individual pace. Questions were solved either by peer-to-peer contact or by asking the course instructor. On the second day, users were familiarized with the XCode integrated development environment to design a graphical user interface for the app. Afterwards, exercises blended interface design with coding to create simple but functional interactive apps. All exercises used healthcare-related content. Apps could be run using XCode's built-in iOS simulator. Participants were invited to bring their personal iOS-based hardware (iPhone, iPad) to run apps on their device.

Evaluation

Participants were requested to answer three questions comparing the situation before and after they attended the workshop. Questions were answered in retrospect by using

a numeric rating scale from 0 to 5, in which 0 was the worst possible result and 5 the best possible result. The questions were:

- A. Rate your general programming knowledge,
- B. Rate your app building skills, and
- C. How important do you consider programming skills.

Another question focused on the legal aspects of medical apps, comparing how many participants would be willing to develop apps dependent on whether they would be supported with legal aspects of software development. A final question was if participants would recommend the workshop to colleagues with the option to leave any comment as free text. A descriptive statistical analysis was performed with IPython in a Jupyter notebook [7].

Results

Demographics

Half of the participants were male (n=7) and half were female (n=7). The majority of participants were between 31-40 years of age (43%), with 51-60 years of age being the second largest age category (29%). Regarding their professional background, the majority of the participants (36%) were categorized as "management", with clinical staff being the second largest category (21% medical and 14% nurse).

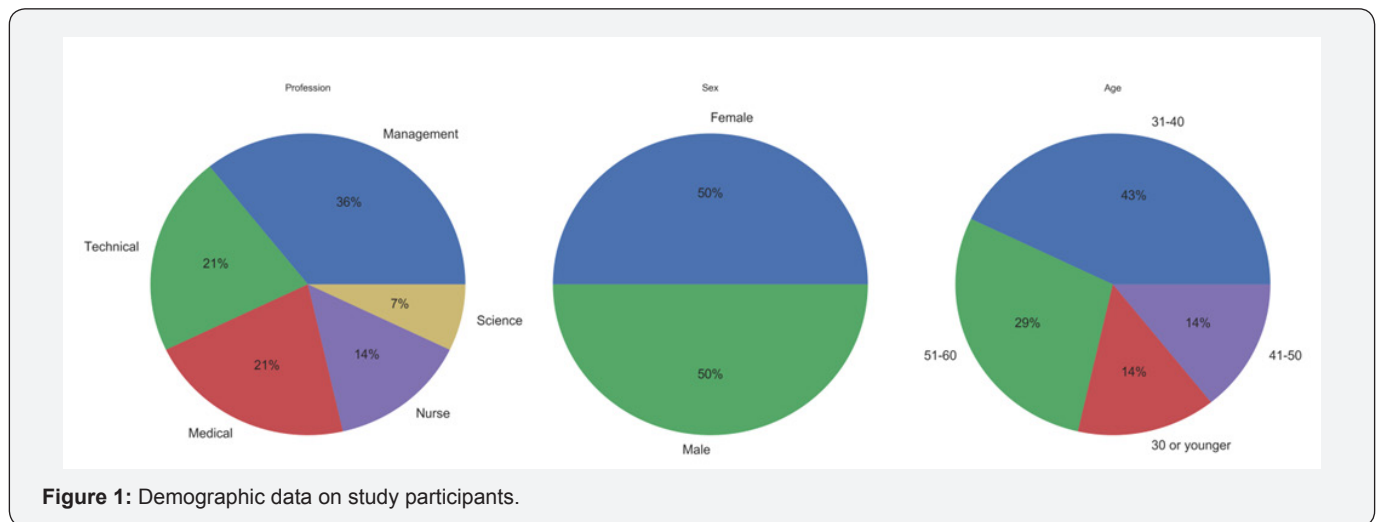


Figure 1: Demographic data on study participants.

Evaluation

Mean and median survey results including their distribution (standard deviation and interquartile range (IQR)) are displayed in Table 1 and differences between pre-workshop and post-workshop are displayed in Table 2. Given the small sample size, no statistical hypothesis testing was performed. General programming knowledge improved with a mean of 1.4±0.5 and a median of 1.0 (IQR 1.0) points. App building skills improved with

a median of 0.9±0.5 and a median of 1.0 (IQR 0.0) points. The degree to which participants considered having programming skills to be important increased with a mean of 1.2±0.9 and a median of 1.0 (IQR 2.0) points. The willingness to develop apps if participants were to be supported in legal aspects of software development increased with 1.0±0.7 and a median of 1.0 (IQR 0.0) points. Finally, all participants would recommend the course to colleagues. Figure 2 visualizes the first four items as a boxplot.

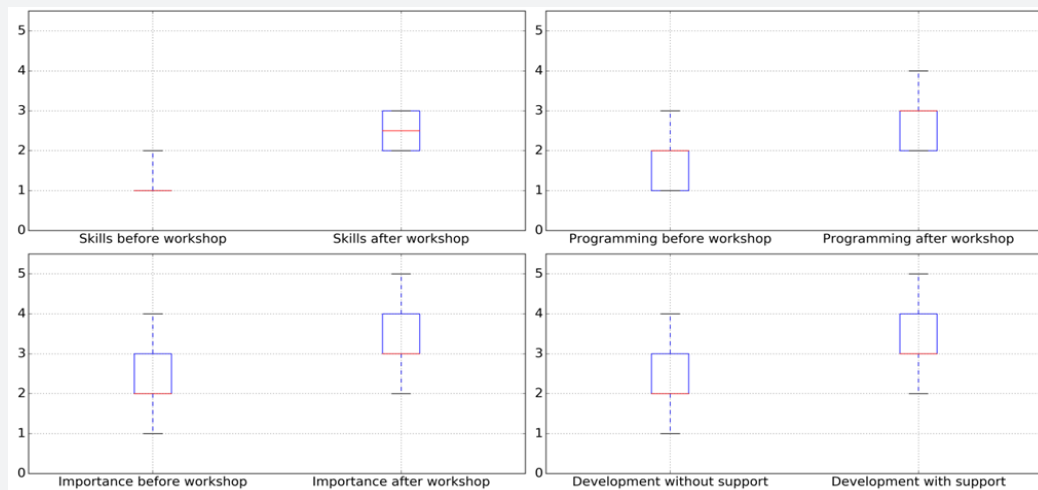


Figure 2: Comparison before and after workshop.

Table 1: Mean and median survey results (including distribution).

	Mean	Mean Std	Median	Median IQR
General programming knowledge before workshop	1.1	0.4	1.0	0.0
General programming knowledge after workshop	2.5	0.5	2.5	1.0
App building skills before workshop	1.8	0.8	2.0	1.0
App building skills after workshop	2.7	0.7	3.0	1.0
Importance of programming skills before workshop	2.3	0.9	2.0	1.0
Importance of programming skills after workshop	3.5	0.9	3.0	1.0
Continue programming without hospital support	2.3	0.9	2.0	1.0
Continue programming with hospital support	3.3	0.9	3.0	1.0

Abbreviations: IQR=Interquartile range; std= Standard Deviation

Table 2: Mean and median improvements after training (including distribution).

	Mean Improved	Mean Improved Std	Median Improved	Median Improved Iqr
General programming knowledge	1.4	0.5	1.0	1.0
App building skills	0.9	0.5	1.0	0.0
Importance of programming skills	1.2	0.9	1.0	2.0
Continue programming with vs. without hospital support	1.0	0.7	1.0	0.0

Abbreviations: IQR=Interquartile Range; std= Standard Deviation

Discussion

In many professions it is mandatory to have digital skills. Currently standard digital skills mainly refer to handling internet, email and text processing. Our society is increasingly becoming an information society and healthcare is no exception. New skills are required to keep up with the huge amount of data overloading hospital staff. First, data should be translated into information. Second, information should be applied in clinical practice. The upcoming field of (clinical) data science combines content expertise with statistical/mathematical skills and computer science to generate new insights from (huge amounts of) data. Big data analysis requires a different approach than regular statistics taught during medical education. Machine learning expands the standard statistical toolkit available to learn from data and produce predictive analytics. Web sites and mobile applications can help to implement all such information in clinical practice. In order to actively participate in this new and exciting field, standard digital skills as mentioned above are insufficient. Computer programming skills are necessary, and an increasing number of currently available programming languages offer a convenient starting point to acquire these skills. This pilot demonstrates feasibility of teaching such skills to an audience of medical doctors, nurses and hospital staff. Two days of training was sufficient to get a basic level of skills to build a simple mobile app. Participants expressed a desire to continue using these skills, provided they would get support from the

hospital such as a helpdesk for questions and regulatory affairs. The limitation of this survey is that participants performed a subjective self-assessment. For this pilot study we consider this to be sufficient as we were most interested in attitude change en demystifying computer programming. However, this approach is unreliable to provide objective outcomes.

Programming languages

In this pilot study we used Swift as a programming language [5]. It has been introduced by Apple in 2014 to gradually replace Objective C, and its syntax offers many similarities with Python which is a widely used introductory programming language. Many other programming languages exist and the fundamentals of programming are present in all languages (e.g. calculations, conditional statements and repetitive statements). Once past these fundamentals, differences exist that should be taken into account when choosing a programming language. Most medical doctors, nurses and other clinical hospital staff will not learn a programming language for the sake of learning the fundamentals, but to build an actual application to help them solve a problem. Table 3 provides an overview of different programming languages. This overview is not meant to be exhaustive and we acknowledge that categorizing parameters as present or absent may be an oversimplification. Also, the level of difficulty is subjective, but a reasonable estimation. Still, from a caregiver's perspective it is useful to have some directions when choosing to learn how to program.

Table 3: Overview of programming languages.

Language	PC Platforms	GUI	Web	Mobile App	Statistics / ML	Open Source	License	Level
C#	W,M,L	+	+ (ASP.NET)	hybrid	+	+ (.NET Core)	Free	Moderate
C++	W,M,L	+	-	hybrid	+	+	Free	Advanced
Java	W,M,L	+	+	Android	+	+	Free	Moderate
JavaScript	W,M,L	web	+	hybrid	-	+	Free	Beginner
Mathematica	W,M,L	±	-	-	+	-	Commercial	Moderate
MATLAB	W,M,L	±	-	-	+	-	Commercial	Moderate
Objective C	M	+	-	iOS	-	-	Free	Moderate
PHP	W,M,L	web	+	-	-	+	Free	Beginner
Python	W,M,L	±	+ (Django)	-	+	+	Free	Beginner
R	W,M,L	±	-	-	+	+	Free	Moderate
Ruby	W,M,L	+	+ (Ruby on Rails)	hybrid	-	+	Free	Beginner
Swift	M	+	-	iOS	-	+	Free	Beginner

Abbreviations: GUI=Graphical User Interface; L=Linux OS; M=Mac OS X; ML=Machine Learning; W=Microsoft® Windows

For programs that are required to run on desktop and laptop, the best options are C#, Java Swift. C# is focused on Microsoft Windows using the .NET framework, but using Xamarin Studio it can also be used for Mac OS X and Linux programs. Java is truly cross-platform but requires a Java virtual machine to make programs run. Swift can be used for native Mac OS X apps besides iOS mobile apps. Although many other languages can also be used to develop desktop programs with a graphical user

interface (GUI), their focus is not on GUI applications and for that reason they may not be the choice of the beginning programmer.

For websites, the most straightforward approach is using the combination of Hypertext Markup Language (HTML), Cascading Stylesheets (CSS) and JavaScript (client-side scripting). As HTML and CSS are markup languages but no programming languages, they are left out of Table 3. PHP is a common addition for server-

side scripting, often in combination with MySQL as an open-source database solution. Alternative traditional solutions for web development are C# in combination with ASP.NET, or Java. Server requirements differ dependent on the web programming language used. Ruby is a modern scripting language which is used a lot for web development in combination with the Ruby on Rails framework, and also suitable for beginners.

For mobile apps there is a choice between native, web and cross platform hybrid apps. Native apps are developed using the mobile operating system's preferred programming language, which is Objective-C or Swift for iOS and Java for Android. This approach offers the best access to the operating system of the target devices and their surrounding ecosystem (e.g. a smartwatch), but targets either iOS or Android. Cross platform hybrid apps are developed in one language targeting both iOS and Android, but need conversion to native binaries by an intermediate layer. The web and hybrid approaches offer the fastest approach to target both iOS and Android (and sometimes even more platforms) but needs to compromise on device-specific functionality and quality of user experience to some extent [8]. Teaching clinicians how to program, using a web based in contrast to our native approach, has been attempted before [9].

For specific statistics or machine learning applications, Python or R may be the best choice for the beginning programmer. Mathematica and MATLAB can offer this functionality too, but they require a commercial license. In addition to the previously mentioned functionality, these packages offer additional mathematical and engineering features. An important additional consideration when choosing a computer programming language is the available support. A language that is widely used will have more extensive documentation and online resources, and questions will be solved sooner. All languages mentioned in Table 3 have a sufficiently large user base to get up to speed easily. On purpose, differences in programming design (object oriented, functional, procedural) have been left out of scope. For the beginning programmer who has the desire to build relatively straightforward small applications, we consider these not to be

relevant. Regulatory affairs have been touched briefly during the course. The purpose of the course was to give an introduction to programming, but as the focus was explicitly on medical apps a short explanation of medical device directives and CE (Conformité Européenne) marking has been provided. Also, raising awareness regarding privacy issues was given particular attention without going in detail.

Conclusion

Teaching computer programming to medical doctors, nurses and hospital staff is feasible and an introductory level of skills can be taught in a couple of days. This audience can benefit from these skills in several areas: desktop applications, web sites, mobile apps and specific statistical or machine learning programs. The preferred programming language to learn may differ dependent on the area of this interest. We discussed directions and considerations to take into account for this particular audience when starting to learn computer programming. More research is needed on the particular needs to teach programming skills to this population, who demonstrated to be willing to learn such skills provided adequate support is available from the hospital.

References

1. <http://codeweek.eu/>
2. <https://csedweek.org/>
3. <https://hourofcode.com/>
4. Zhang MW, Tsang T, Cheow E, Ho CSh, Yeong NB, et al. (2014) Enabling Psychiatrists to be Mobile Phone App Developers: Insights Into App Development Methodologies. *JMIR mHealth uHealth* 2(4): e53.
5. <https://developer.apple.com/xcode/>
6. <https://developer.apple.com/swift/>
7. <https://github.com/DigNeurosurgeon/intro-med-apps/blob/master/IntroMedAppsExcel.ipynb>
8. Dalmasso I, Datta SK, Bonnet C, Nikaein N (2013) Survey, comparison and evaluation of cross platform mobile application development tools, 2013 9th Int Wirel Commun Mob Comput Conf, pp. 323-328.
9. Subhi Y, Todsén T, Ringsted C, Konge L (2014) Designing web-apps for smartphones can be easy as making slideshow presentations. *BMC Res Notes* 7: 94.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/OAJNN.2017.04.555632](https://doi.org/10.19080/OAJNN.2017.04.555632)

Your next submission with Juniper Publishers will reach you the below assets

- Quality Editorial service
 - Swift Peer Review
 - Reprints availability
 - E-prints Service
 - Manuscript Podcast for convenient understanding
 - Global attainment for your research
 - Manuscript accessibility in different formats
- (Pdf, E-pub, Full Text, Audio)**
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>