

Editorial

NeuroMind: Past, present, and future

Pieter L. Kubben

Department of Neurosurgery, Maastricht University Medical Center, The Netherlands

E-mail: *Pieter L. Kubben - pieter@kubben.nl

*Corresponding author

Received: 03 April 17 Accepted: 28 June 17 Published: 06 September 17

Abstract

This narrative report describes the underlying rationale and technical developments of NeuroMind, a mobile clinical decision support system for neurosurgery. From the perspective of a neurosurgeon – (app) developer it explains how technical progress has shaped the world’s “most rated and highest rated” neurosurgical mobile application, with particular attention for operating system diversity on mobile hardware, cookbook medicine, regulatory affairs (in particular regarding software as a medical device), and new developments in the field of clinical data science, machine learning, and predictive analytics. Finally, the concept of “computational neurosurgery” is introduced as a vehicle to reach new horizons in neurosurgery.

Key Words: Clinical data science, machine learning, mhealth, mobile, predictive analytics, programming

Access this article online**Website:**www.surgicalneurologyint.com**DOI:**

10.4103/sni.sni_129_17

Quick Response Code:**INTRODUCTION**

During my medical training I started developing websites after being inspired by a colleague who used to do so. I bought a book on web development, and started with static content using only HTML (Hypertext Markup Language). Later this evolved into interactive content. I created an educational website with interactive patient cases for neurology and did a pilot study on whether open questions were better than multiple choice questions for online assessment (in our results, they were not). Afterwards I moved towards PHP (which stands for PHP Hypertext Processor) and MySQL (SQL stands for Structured Query Language, and MySQL is the most known open source approach) for more advanced database-driven interactivity. At that moment PDA’s (Personal Digital Assistants) were hitting the market, and back then you had to choose between Palm OS (Palm Inc, Sunnyvale, CA) and Pocket PC (Microsoft, Redmond, WA). There were some third party solutions for cross-platform development, such as AvantGo.com (San Mateo, CA) and Mobipocket.com (Paris, France). These solutions accepted web pages

to be “clipped” on the device after synchronisation with a desktop computer or laptop using a so-called “cradle” (a docking station for a palmtop device). This approach allowed to access web content offline, which was important as mobile data networks were limited and relatively expensive. Also, wireless networking was not available on the first mobile devices. In 2003 I started working as a resident in neurosurgery and soon I realized there were many grading scales and classification systems that I needed to apply in my clinical work but –as I did not need to apply them on a regular basis - I had trouble remembering them. I built a solution using PHP, MySQL, and Mobipocket and improved it throughout the years afterwards. Later, I added some interactivity using

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License, which allows others to remix, tweak, and build upon the work non-commercially, as long as the author is credited and the new creations are licensed under the identical terms.

For reprints contact: reprints@medknow.com

How to cite this article: Kubben PL. NeuroMind: Past, present, and future. *Surg Neurol Int* 2017;8:216.

<http://surgicalneurologyint.com/NeuroMind:-Past,-present,-and-future/>

JavaScript. In contrast to PHP, which is executed at the server (server-sided scripting), JavaScript is executed in the browser (client-sided scripting) and therefore worked well in this clipping approach.

NeuroMind 1.0

After introduction of the iPhone (2007) and the App Store (2008) I immediately loved the device and its refreshing approach towards mobile computing. Obviously, I wanted to have my grading scales and classification systems with me on this new device to support my “palmomental reflex.”^[5] I took up the challenge to learn programming for the iPhone (back then in Objective C) and after some time struggling with Apple’s so-called provisioning profiles (a technical requirement in the app to be allowed in the App Store) the first version of NeuroMind was released early 2010 [Figure 1]. The app contained circa 30 classification systems, some royalty free neuroanatomy images, some checklists, and a few mindmaps. At the same time I started experimenting with blogs and Twitter, which helped to draw attention to the app. As there were a limited amount of medical apps and almost no neurosurgical apps at the time, NeuroMind got quite some visibility on other blogs and social networks. Initial downloads were 2,500 weekly – far beyond my expectation – which led to more interest from other websites and some magazines. When asked about what made the app as successful as it is, I always give the same two reasons: (1) “scratch your own itch” (I had a problem for which I created the app, and apparently more people had that problem); and (2) it surely helped to be early in the App Store and to use some social networks.

NeuroMind 1.5

After introduction of the iPad mid 2010 many apps introduced a so-called “HD” version referring to the larger screen (which was not high definition or “retina” by any means). To create a NeuroMind version for iPad some serious rework was necessary. Meanwhile the Android operating system (Google Inc, Mountain View, CA) was attracting more users, but programming for Android is done in Java – a completely different programming language than Objective C. For that reason I started looking for cross-platform development frameworks targeting both iPhone OS (the formerly iOS) and Android. My framework of choice became Titanium Studio (Appcelerator Inc, San Jose, CA), which uses a dedicated Javascript API (Application Programming Interface) that gets translated into native device code. The advantage of this approach is that the native user interface “look and feel” of each device is maintained, whereas a more “hybrid” web app cannot offer this usability advantage. Late 2010 NeuroMind 1.5 was introduced offering support for iPhone, iPad [Figure 2], and Android [Figure 3]. The code was fully rebuilt in Titanium and another 10 classification systems were added and the mindmaps were removed.



Figure 1: NeuroMind 1.0 on iPhone

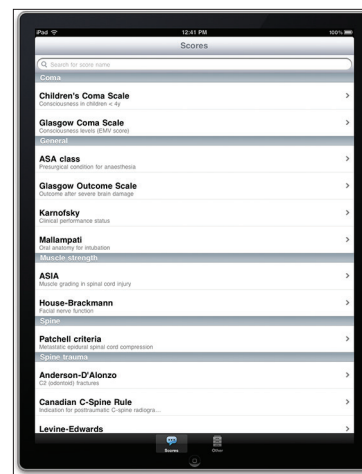


Figure 2: NeuroMind 1.5 on iPad

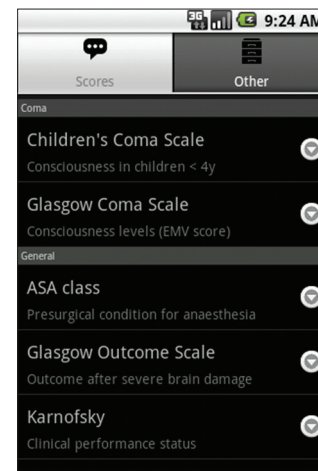


Figure 3: NeuroMind 1.5 on Android

NeuroMind 2.0

The main disadvantage of NeuroMind 1.x was that it remained fully static. While each minor update contained some more classification systems, the format remained text-only and let the end user do the math.

In 2011, I gave a TED talk explaining my thoughts on clinical decision support systems.^[17] In summary, mobile decision support apps should assist the caregiver to arrive quickly at those pieces of information which should be considered for that particular patient at that particular moment. They are not called decision *making* systems for a reason: the final decision making is done by the caregiver and at his or her responsibility. Translating a text-based guideline into a clinical decision support system can be reduced to a two-step process: translate the guideline into a flowchart, and translate the flowchart into programming code. Obviously a flowchart should contain a flow which is unambiguous, so that each step leads to a clear decision and subsequent steps. This is mandatory for a computer program, which needs to receive clear instructions. Some refer to such step-by-step instructions as “cookbook medicine,” which leads to mixed opinions.^[6,7,13,16,18] My usual response is that a cookbook does not prescribe you what to do, it merely suggests what you could do and leaves you free to adapt your decision to the particular circumstances. Which is not any different than what we consider evidence-based medicine to be, and this has my full support.^[15] Early 2012, I released NeuroMind 2.0 for iOS and Android, which contained some interactive guidelines and risk calculators, and some more classification systems [Figure 4]. Mid 2012 I was approached by Intel Corporation (Santa Clara, CA) to know if I was willing to port NeuroMind to Windows 8 as they were preparing the release of this new operating system with a mobile-friendly interface, together with new hardware that would allow users to benefit from this. I built this version in C#, the preferred native language for Windows programming, and adhered to the “tiles” user interface [Figure 5]. NeuroMind for Windows was released late 2012 and contained only the static content, which has not changed to date. Early 2013, the iOS version was updated to be compliant with the larger

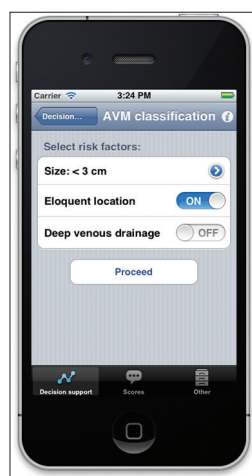


Figure 4: NeuroMind 2.0 on iPhone, introducing interactive decision support

screen size of the iPhone 5, and late 2013 it got another update for the new “flat layout” user interface of iOS 7.

NeuroMind 2.5

In 2014, increased attention was drawn to regulatory affairs for mobile medical applications, in particular focusing on the question when they needed to be considered as a medical device. The FDA released its first non-binding guidance on mobile medical applications back in 2011 with an update in 2013.^[4] In Europe, the Medical Device Directives (MEDDEV) assist in determining whether software has to be considered as a medical device and if it needs CE-marking. As a brief summary, if the app does more than simply storing and displaying information (which may be compared with an ebook or NeuroMind 1.x), especially when some data manipulation or calculation is being performed for a clinical purpose, chances are high that it has to be considered as a medical device. The Dutch Health Inspectorate approached me with some questions on NeuroMind, and (in a supportive manner) helped me realize that NeuroMind should be considered as a class I medical device from a MEDDEV perspective and needed CE-marking. The good thing about being a class I medical device is that CE-marking can be done by the developer and that a so-called notified body is not required (which would make registration much more costly). The two essential questions you need to answer for this process are: how do you minimize the chance of an error getting into the application, and how do you minimize the potential damage if an error exists.

The first question is merely a matter of testing the code and some potential scenarios of a clinical decision support module. Being both the developer and the end user I am fortunate to be able to perform these steps myself. Additionally, all content is referenced in the app so other users can check the resources being used to generate the content (and where applicable, the mindmaps were available on my blog too). The second question is more tricky: it is easy to add some options for users to reach you from within the app in case they find an error or want to ask a question. I prefer e-mail for this, and this has been present ever since NeuroMind 1.0. The main problem though is how to force the end user to update the application if an error exists and the user is not aware of this. NeuroMind has always been an application that could be used completely offline. This goes back to my work on PDA's and the philosophy that content should be available regardless of available network connectivity (many hospitals do not have full wifi coverage or won't allow cellular networks in some departments, like the intensive care units). Holding on to this offline philosophy would introduce many challenges that were not present for an online version. Moreover, an online version would allow me to change content very quickly (updating an online database) and

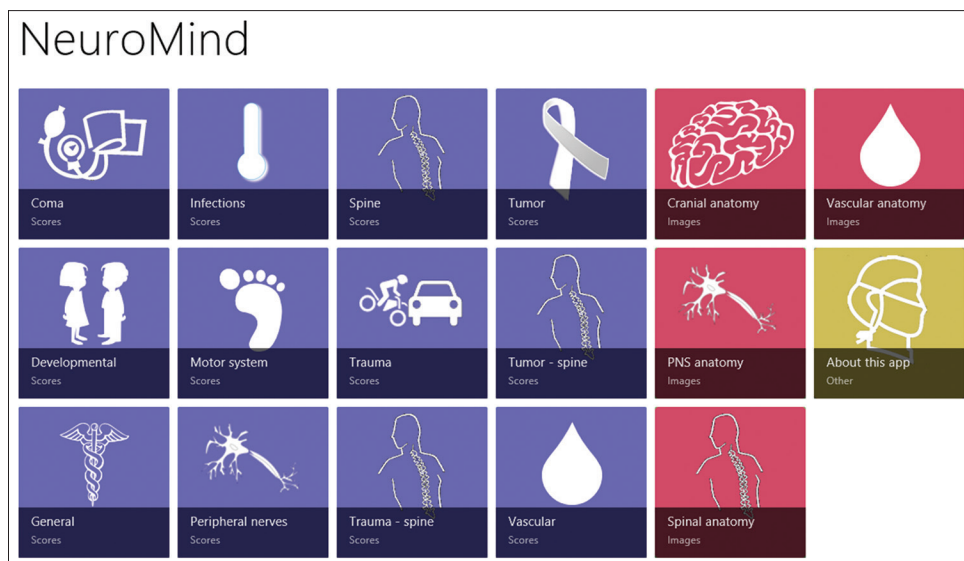


Figure 5: NeuroMind for Windows

the new content would be immediately available on all devices without manual uploads to both the App Store and Google Play. Despite using a cross platform development framework, uploads to the different stores is still a manual process. In case of the App Store, the update's availability in the App Store is delayed a few days (sometimes up to two weeks) due to Apple's internal review process before release. There were very compelling reasons to convert NeuroMind to an online version. The main disadvantage was that I had to redevelop all interactive modules to a web-supported format as the dedicated Javascript API in Titanium had too limited support for an online version. Therefore, I rebuilt the risk calculators that I found most relevant in web technology, relying on jQuery Mobile (jQuery.com, The jQuery Foundation) for the front-end and PHP for the back-end logic of the application. As PHP is a server-sided scripting language, the back-end code is not accessible from the browser and therefore not directly available for anyone who would like to know how the app arrives at a certain decision. For the CE-marking process, I decided to make this code available on GitHub (GitHub Inc, San Francisco, CA). NeuroMind 2.5 for iOS and Android was released late 2014 and CE-marked as a class I medical device [Figure 6]. The CE logo is ever since visible when starting the app, as it is required to have to logo visible on the "package" of the medical device.

New inspiration

By 2015, I started losing motivation for further development of NeuroMind. A sense of "been there, done that" came up frequently and I was wondering how to continue. Meanwhile I started learning the Python programming language in the context of data science, to which I will refer later in this article. Python is relatively easy to pick up and an excellent language to

start learning programming. For this reason I created the website www.programmingforphysicians.com (or www.programmingfordoctors.com) in reply to many email messages I received the past years in which medical students, residents, and some staff members asked me for advice on how to learn programming.^[9,10] Therefore, I also felt attracted to Swift, the new programming language that Apple introduced in 2014 and I started working with in 2015. It's syntax is fairly similar to Python and developing iOS apps with Swift turned out to be much more fun than using Objective C or even the Titanium platform I had been using. Also XCode, which is Apple's IDE (Integrated Development Environment) for iOS and macOS development, had been improved significantly throughout the past years and was now a pleasure to work with. My first app in Swift was a dose calculator for the NASCIS scheme, which was rejected by Apple because drug dosage calculators were only allowed from a university (medical center) or drug company, not from a private developer. My appeal did not help, and the app never made it into the App Store. I open sourced my code on GitHub for others to learn from. Afterwards I built TREMOR12, an open source app for tremor measurement in patients with Parkinson's disease and essential tremor.^[12] It measures acceleration, rotation, rotation speed, and the device's orientation during measurements and –in contrast to other tremor apps- exports all raw data as a CSV (comma separated values) file for further processing. The app was accompanied by an open source tremor processing module written in Python.

Having found back the joy in iOS development I decided to rebuilt NeuroMind from scratch, using Swift and XCode. The obvious side effect would be that Android users would not benefit from this update,

but for two reasons I did not mind at the time. First, the download figures suggested that there were many more iOS users than Android users for NeuroMind. Second, I decided to offer some In App Purchases to cover for my development-related expenses and future investments. This was easier to do when focusing on one platform.

NeuroMind 3.0

Early 2016, NeuroMind 3.0 for iOS was released [Figure 7]. It included new features like optimal iOS integration, favorites, a master-detail view on iPad (which gives the side menu like in the Mail application) and export of decision support data as a CSV (comma separated values) file for administration and research purposes. Two decision support modules were available as In App Purchase, all static content was free of charge. It took a while before Apple approved this update, and there were a few rejections for mentioning that NeuroMind was also available for Android and Windows on several places (seriously). All decision support modules needed to be reprogrammed in Swift and it turned out that there was an error in one of them. The remote notification system that I developed to remain CE compatible did not work as good as it worked during testing, and it took a few days before Apple approved the minor content update, causing me some stress. Further, although In App Purchases were sufficient to cover my expenses, I did not feel comfortable with this approach in combination with some disaffection for Apple's review policy, which seemed to be more focused on optimizing their market advantage than ensuring content quality. Lastly, using social networks for some polls, Android usage turned out to be higher than expected based on downloads, and the Android version was getting behind with regard to the technical aspects (last updated in 2014). Time for another make-over which led to the current version.

NeuroMind.cc

NeuroMind.cc is the current version of NeuroMind, and is available on all devices and all screen sizes. All content is available free of charge. Moreover, it is also directly accessible from any modern web browser on desktop or laptop using the weblink <http://neuromind.cc> [Figure 8]. The abbreviation "cc" stands for "continuous" and "collaborative". Continuous refers to the fact that it is accessible from any device, regardless of any app store's policy or your personal preferences for the hardware and software you like to work with. Collaborative refers to the opportunity to comment directly on classification systems and decision support modules from within the app. There is no need to register to be able to comment. Technically, NeuroMind.cc is backed by WordPress.org and an Ionic-based responsive front-end by Appresser.com, and customized by some theme style changes and dedicated PHP code. The binary files for the App Store and Google Play are generated on Phonegap.com. This means that

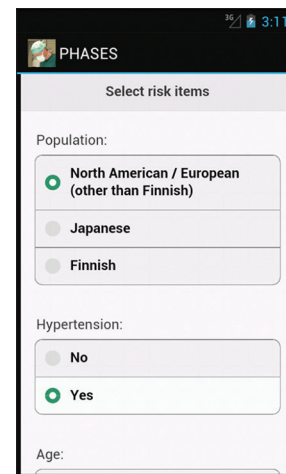


Figure 6: NeuroMind 2.5 on Android (introducing CE mark)

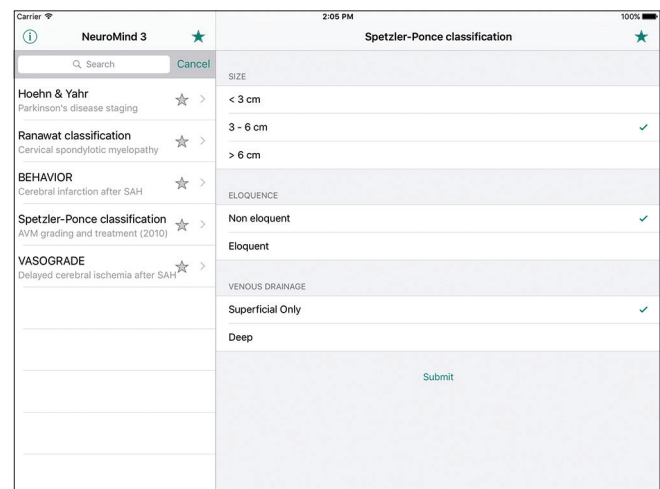


Figure 7: NeuroMind 3.0 on iPad

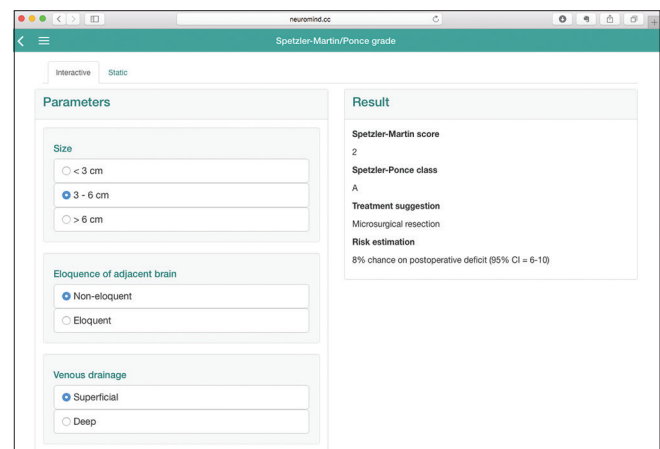


Figure 8: NeuroMind.cc from the web browser

NeuroMind.cc requires web access, unlike NeuroMind 3.0. There are many maintenance advantages to this approach, most importantly that users always have the latest version (which is very important related for safety reasons) and that the content can be built using web standards.

The Wordpress back-end allows easy and quick updates from the browser, and allows for collaborating on content too. A side advantage is that not only download statistics are available (after 300,000 I stopped counting as download numbers do not mean much), but also usage statistics are available. For this purpose Google Analytics is being used (Google Inc, Mountain View, CA), which provides details on which pages have the most views, stratified for geolocation, time, browser, operating system, and more.

Future

Three directions are important for the future of NeuroMind: content, technical, and regulatory. Regarding content, my intention is to add more decision support modules. The advantage of running Google Analytics is that I can view which content is being used most, and hence could benefit of more support and interactivity. Such content could be developed by me, but also by others who wish to contribute to NeuroMind. Current interactive content is based on simple calculations or flowcharts. Therefore, all content flow can be derived from studying the source code. In contrast to this so-called “white box” approach, further technical development may also allow for “black box” approaches in which machine learning techniques are applied for predictive modelling. This field is also referred to as (clinical) data science. The current platform would easily allow to integrate such content, that may be generated in Python or R.^[14] In particular, the latter is widely used in academia for statistical testing, and also an excellent environment for statistical learning (i.e. machine learning, predictive analytics). The integration of shiny^[2] and R Markdown^[1] enables to create responsive webbased front-ends for predictive applications that can benefit from the latest statistical learning techniques. Regarding regulatory affairs, both increased focus on safety with respect to “software as a medical device” and privacy concerns will shape borders for what can and cannot be done.

Computational neurosurgery

Besides an overview of NeuroMind development, underlying philosophy and practical considerations of the world’s “most rated and highest rated” neurosurgical mobile application,^[8] the “best neurosurgical clinical tool,”^[19] this review serves another purpose. It aims to introduce a practical concept of “computational neurosurgery.” Computer programming is an important skill for modern scientists in engineering and other technical fields.^[3] Neurosurgery, being a technical discipline in the field of medicine, would benefit from context experts who do have some programming

skills and help to push the field forward.^[11] I am not a brilliant programmer by any means. I like programming a lot, and have a firm grasp on the basics. This serves as the foundation on which I can build and apply more task-specific knowledge. The story described in this article demonstrates what can be done using this approach from the perspective of a neurosurgeon – (app) developer. I sincerely hope that this story will inspire others – students, residents, staff neurosurgeons – to learn the fundamentals of programming and clinical data science, and apply it for the good of neurosurgery.

REFERENCES

- Allaire JJ, Cheng J, Xie Y, McPherson J, Allen J, Wickham H, et al. rmarkdown: Dynamic Documents for R. Available: <http://rmarkdown.rstudio.com>. [Last accessed on 27 October 2016].
- Chang W, Cheng J, Allaire JJ, Xie Y, McPherson J. shiny: Web Application Framework for R. Available: <https://CRAN.R-project.org/package=shiny>. [Last accessed on 27 October 2016].
- CodeWeek EU. CodeWeek EU. <http://codeweek.eu>. [Last accessed on 27 October 2016].
- Cortez NG, Cohen IG, Kesselheim AS. FDA regulation of mobile health technologies. *N Engl J Med* 2014;371:372-9.
- Crelinsten GL. The intern’s palmomental reflex. *N Engl J Med* 2004;350:1059.
- Dans PE. Credibility, cookbook medicine, and common sense: Guidelines and the college. *Ann Intern Med* 1994;120:966-8.
- Farquhar DR. Recipes or roadmaps? Instead of rejecting clinical practice guidelines as “cookbook” solutions, could physicians use them as roadmaps for the journey of patient care? *CMAJ* 1997;157:403-4.
- Khan NR, Auschwitz TS, Choudri AF, Klimo P Jr. iPhone Resources in Neurosurgery. 14:22–23 Available: https://www.cns.org/system/files/congress_quarterly/CNSQ_13fall_1.pdf. Last accessed on 27 October 2016].
- Kubben PL. Programming for physicians: A crash course. *Surg Neurol Int* 2015;6:15.
- Kubben PL. Programming for physicians: A free online course. *Surg Neurol Int* 2016;7:29.
- Kubben PL. Why physicians might want to learn computer programming. *Surg Neurol Int* 2013;4:30.
- Kubben PL, Kuijff ML, Ackermans LPCM, Leentjes AFG, Temel Y. TREMOR12: An Open-Source Mobile App for Tremor Quantification. *Stereotact Funct Neurosurg* 2016;94:182-6.
- Latoszek-Berendsen A, Tange H, van den Herik HJ, Hasman A. From clinical practice guidelines to computer-interpretable guidelines. A literature overview. *Methods Inf Med* 2010;49:550-70.
- R Core Team. R: A language and environment for statistical computing. Available: <https://www.R-project.org/>. [Last accessed on 27 October 2016].
- Sackett DL, Rosenberg WM, Gray JA, Haynes RB, Richardson WS. Evidence based medicine: What it is and what it isn’t. *BMJ* 1996;312:71-2.
- Steinberg KE. Cookbook medicine: Recipe for disaster? *J Am Med Dir Assoc* 2006;7:470-2.
- TEDxMaastricht - Pieter Kubben - “From medical knowledge to practical healthcare”: TEDxMaastricht - Pieter Kubben - “From medical knowledge to practical healthcare.” Available: <https://www.youtube.com/watch?v=9sNqgqYkBnl>.
- Timmermans S, Mauck A. The promises and pitfalls of evidence-based medicine. *Health Affairs (Project Hope)* 2005;24:18-28.
- Zaki M, Drazin D. Smartphone use in neurosurgery? APP-solutely! *Surg Neurol Int* 2014;5:113-8.