# The Art of Computer Programming – A difficult book to understand

**Donald Knuth**

**Dennis Ritchie**

B.Sc.(Hons) in Software Development

April 12, 2016

**Final Year Project**

Advised by: Dr Alan Turing

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)

# Contents

# About this project

**Abstract** A brief description of what the project is, in about two-hundred and fifty words.

**Authors** Explain here who the authors are.

# Chapter 1

# Introduction

It's a booking web application build using MEAN stack (MongoDB, Express, Angular, Node).The application is made of two websites bookAroom-user and bookAroom-admin. BookAroom-user is using Auth0 authentication system which allows for convenient log in with gmail fb as well as sign up form with password recovery. Once user is signed in he/she can book search and sort workbenches. BookAroom-admin is using json web-tokens identification system. It lets administrative user to add, edit and deletes existing and new workbenches.

# Chapter 2

# Context

- Provide a context for your project.
- Set out the objectives of the project
- Briefly list each chapter / section and provide a 1-2 line description of what each section contains.
- List the resource URL (GitHub address) for the project and provide a brief list of the main elements at the URL.

## 2.1 Filler

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam mi enim, interdum ut elit lobortis, bibendum tempus diam. Etiam turpis ex, viverra tristique finibus nec, feugiat at metus. Curabitur tempus gravida interdum. Donec ac felis a lorem scelerisque elementum. Vestibulum sit amet gravida tortor, a iaculis orci. Nam a molestie augue. Curabitur malesuada odio at mattis molestie. In hac habitasse platea dictumst. Donec eu lectus eget risus hendrerit euismod nec at orci. Praesent porttitor aliquam diam, eu vestibulum nisl sollicitudin vel. Nullam sed egestas mi.

Quisque vel erat a justo volutpat auctor a nec odio. Sed rhoncus augue sit amet nisl tincidunt, vitae cursus tellus efficitur. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque et auctor dui. Fusce ornare odio ipsum, et laoreet mi molestie sed. Cras at massa sit amet ipsum gravida aliquam. Nulla suscipit porta imperdiet. Fusce eros neque, bibendum sit amet consequat non, pulvinar quis ipsum.

### 2.1.1 More filler

Donec fermentum sapien ac rhoncus egestas. Nullam condimentum condimentum eros sit amet semper. Nam maximus condimentum ligula. Praesent faucibus in nisi vitae tempus. Sed pellentesque eleifend ante, ac malesuada nibh dapibus nec. Phasellus nisi erat, pulvinar vel sagittis sed, auctor et magna. Quisque finibus augue elit, consequat dignissim purus mollis nec. Duis ultricies euismod tortor, nec sodales libero pellentesque et. Interdum et malesuada fames ac ante ipsum primis in faucibus.

Donec id interdum felis, in semper lacus. Mauris volutpat justo at ex dignissim, sit amet viverra massa pellentesque. Suspendisse potenti. Praesent sit amet ipsum non nibh eleifend pretium. In pretium sapien quam, nec pretium leo consequat nec. Pellentesque non dui lacus. Aenean sed massa lacinia, vehicula ante et, sagittis leo. Sed nec nisl ac tellus scelerisque consequat. Ut arcu metus, eleifend rhoncus sapien sed, consequat tincidunt erat. Cras ut vulputate ipsum.

Curabitur et efficitur augue. Proin condimentum ultrices facilisis. Mauris nisi ante, ultrices sed libero eget, ultrices malesuada augue. Morbi libero magna, faucibus in nunc vitae, ultricies efficitur nisl. Donec eleifend elementum massa, sed eleifend velit aliquet gravida. In ac mattis est, quis sodales neque. Etiam finibus quis tortor eu consequat. Nullam condimentum est eget pulvinar ultricies. Suspendisse ut maximus quam, sed rhoncus urna.

## 2.2 Filler

Phasellus eu tellus tristique nulla porttitor convallis. Vestibulum ac est eget diam mollis consectetur. Donec egestas facilisis consectetur. Donec magna orci, dignissim vel sem quis, efficitur condimentum felis. Donec mollis leo a nulla imperdiet, in bibendum augue varius. Quisque molestie massa enim, vitae ornare lacus imperdiet non. Donec et ipsum id ante imperdiet mollis. Nullam est est, euismod sit amet cursus a, feugiat a lectus. Integer sed mauris dolor.

Mauris blandit neque tortor, consequat aliquam nisi aliquam vitae. Integer urna dolor, fermentum ut iaculis ut, semper eu lacus. Curabitur mollis at lectus at venenatis. Donec fringilla diam ac risus imperdiet suscipit. Aliquam convallis quam vitae turpis interdum, quis pharetra lacus tincidunt. Nam dictum maximus lectus, vitae faucibus ante. Morbi accumsan velit nec massa tincidunt porttitor. Nullam gravida at justo id viverra. Mauris ante nulla, eleifend vitae sem vitae, porttitor lobortis eros.

Cras tincidunt elit id nisi aliquam, id convallis ex bibendum. Sed vel

odio fringilla, congue leo quis, aliquam metus. Nunc tempor vehicula lorem eu ultrices. Curabitur at libero luctus, gravida lectus sed, viverra mi. Cras ultrices aliquet elementum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Sed metus ante, suscipit sit amet finibus ut, gravida et orci. Nunc est odio, luctus quis diam in, porta molestie magna. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris pulvinar lacus odio, luctus tincidunt magna auctor ut. Ut fermentum nisl rhoncus, tempus nulla eget, faucibus tortor. Suspendisse eu ex nec nunc mollis pulvinar. Nunc luctus tempus tellus eleifend porta. Nulla scelerisque porttitor turpis porttitor mollis.

Duis elementum efficitur auctor. Nam nisi nulla, fermentum sed arcu vel, posuere semper dui. Fusce ac imperdiet felis. Aenean quis vestibulum nisl. Integer sit amet tristique neque, at suscipit tortor. Morbi et placerat ante, vel molestie dui. Vivamus in nibh eget massa facilisis accumsan. Nunc et purus ac urna fermentum ultrices eget sit amet justo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Cras elementum dui nunc, ac tempor odio semper et. Ut est ipsum, sollicitudin eleifend nisl eu, scelerisque cursus nunc. Nam at lectus vulputate, volutpat tellus vel, pharetra mauris. Integer at aliquam massa, at iaculis sem. Morbi nec imperdiet odio. In hac habitasse platea dictumst.

Mauris a neque lobortis, venenatis erat ut, eleifend quam. Nullam tincidunt tellus quis ligula bibendum, a malesuada erat gravida. Phasellus eget tellus non risus tincidunt sagittis condimentum quis enim. Donec feugiat sapien sit amet tincidunt fringilla. Vivamus in urna accumsan, vehicula sem in, sodales mauris. Aenean odio eros, tristique non varius id, tincidunt et neque. Maecenas tempor, ipsum et sollicitudin rhoncus, nibh eros tempus dolor, vitae dictum justo massa in eros. Proin nec lorem urna. In ullamcorper vitae felis sit amet tincidunt. Maecenas consectetur iaculis est, eu finibus mi scelerisque et. Nulla id ex varius, ultrices eros nec, luctus est. Aenean ac ex eget dui pretium mattis. Ut vitae nunc lectus. Proin suscipit risus eget ligula sollicitudin vulputate et id lectus.

# Chapter 3

# Methodology

About one to two Page Describe the way you went about your project:

- Agile / incremental and iterative approach to development. Planning, meetings.
- What about validation and testing? Junit or some other framework.
- If team based, did you use GitHub during the development process.
- Selection criteria for algorithms, languages, platforms and technologies.

# Chapter 4

# Technology Review

## 4.1 Introduction

About seven to ten pages.

Developing native applications separately for each platform is a laborious and expensive undertaking [1]. There has been an enormous increase in different types of mobile devices, operating systems and screen sizes; which creates issues from an application development point of view, when developing an application for each device. If you only take into account major operating systems out there like Android, iOS and Windows, that's three separate platforms to develop for and if you want to target all of the market it would include others like Symbian Nokia, Blackberry SIM, Samsung BADA, WebOS and HP that's 8 platforms to target 99 percent of the market. This literature review analysis reviews and supports four different categories of frameworks for mobile cross platform development that has been created in response of heterogeneity of mobile devices by referencing papers from the ACM and IEEE. It describes trends and issues of each type and which category and framework is leading in mobile cross platform development.

## 4.2 Existing and Emerging Operating Systems for Mobile Devices

There are four dominant mobile operating systems according to the IDC 2015 Report [2]. Google's Android (82.8Apple's iOS (13.9Blackberry (0.3Sailfish and Ubuntu Touch OS but they only account for 0.4While Android is the dominant operating system, it is supported on multiple 2 platforms and it is in sync with their vision of "write once, run everywhere" [3]. There are

other emerging mobile devices with their own operating systems; for instance Mozilla launched ZTE in early 2013 [4] but its market share hasn't grown and is still under 0.4two of the biggest players in the market are iOS and Android [5]. They both have communities built around them, are well matured and have been thoroughly tested. There are many open source libraries, especially for Android. As evidence suggests Android has the biggest market share. Over the years Android has kept growing in market share, acquiring bigger market share each year and predictions have been made that it will stop or slow down; the Yankee Group forecast stated in Analysis of Cross-Platform Development [21].

## 4.3 Categories and Frameworks of Cross Platform Development

Categories and Frameworks of Cross Platform Development In todays world, application development for smart devices is an evolving field with great economic and scientific interest [6]. Cross platform is a wide area of development and according to Spyros Xanthopouls and Stelios Xinnogalas' paper [6], it has split into four main categories: Interpreted applications, hybrid applications, generated applications and web applications. Each category containing its own framework. The reason for this is that the user experience for each framework creates a certain level of native look and feel for an application against the actual operating system; running generated applications achieves the highest native look because it's compiled in the OS' native language. Web applications have the lowest native look and feel since they are trying to target all markets using web technologies (HTML5, CSS, JS). There are various frameworks emerging, to name a few according to Oliver Le Goaer and Sacha Waltham's paper [7]: Rhodes, LiveCode, PhoneGap, Titanium, Tabris, Neomades XMLVM, Canappi, APPlause, MoSync SDK, Codename One and Marmalade SDK. Few are more matured then others and some well supported frameworks are Titanium and PhoneGap. The problem that Cross-Platform is trying to solve is well described in paper [9] and it indicates as follows: 3 "A mobile application may require to be developed several times, one for each supported platform, thus dramatically increasing the required time and skills for developers, and finally, the cost of production. A solution is represented by Framework for cross platform development."

## 4.4 Interpreted Applications

These applications are interpreted natively to a platform that it's running on, so if it's running on Android it will be interpreted in Java and if running on iOS it will be Objective-C, thus creating a one hundred percent native look and feel in the application. Other benefits include being efficient and running smoothly; since the app is interpreted natively. One of the new frameworks for interpreted applications is MD 2[13]. One of the more mature frameworks would be Titanium which is discussed in more detail below.

### 4.4.1 Titanium

One of its most valuable features is that is is open source and has been in development since 2006. It was released in 2008 by the Appcelerator company and is a commercially supported product with its source code released under the Apache 2 license [8]. It can be installed on multiple operating systems including Windows, Macintosh and Linux. It allows the creation of Android, iOS, Windows Phone, BlackBerry OS and Tizen apps from a single code base. It's based on the MVC pattern (Model View Controller) so it's a loosely coupled stable design. Only one JavaScript language is now required to develop with this framework; this makes it easy to learn and powerful, you can program all three aspects of your application: Model, View and Controller in just one language and with the new JavaScript ECMAScript standard 6 coming out this should make it even more robust. It allows the use of Cloud (Server) services as a ready to use mobile backend. When the Titanium application is compiled the engine processes the JavaScript and builds the appropriate native application for the specific platforms (iOS uses Objective-C, Android uses Java) thus ensuring a native look and feel for the application. One of the downsides is that support is limited on iOS and Android [9]. The advantage of this framework is the ability to compile the application with the native APIs; it provides a wider set of native device functionalities that web applications cannot provide [10].

## 4.5 Hybrid Applications

Hybrid applications are primarily built using HTML5, they behave like a website. The way they work is that they have a container within the target platform, named UIWebView in iOS and WebView in Android; this allows them to use phone features such as the accelerometer, GPS and camera. These new capabilities were able to be implemented with the advent of

HTML5. The user interface is not generated natively as previous seen in interpreted applications, so technology like jQueryMobile and Sencha Touch [12] are used to achieve native effect but they are not one hundred per cent native. One of the more popular frameworks is PhoneGap which is based on top of Cordova, described in more detail below.

### 4.5.1 PhoneGap

PhoneGap which is based on top of Apache Cordova was created by a company Nitoby and is supported by Adobe Systems; it is open source. The programmer needs to know HTML, CSS and JavaScript to develop applications using this framework. PhoneGap allows deployments to many different platforms including Android, iOS, Windows Phone, BlackBerry, WebOS Symbian and Bada [11]. PhoneGap is a hybrid application because it's not pure HTML/JavaScript. PhoneGap has a bridge between target platforms such as Android, iOS and Windows Phone which connects the JavaScript API to the target platform device and allows developers to use such features as the camera, accelerometer, network, storage and others. The place where this framework is lacking is that the developer has to create his own style sheets to make the application feel native to its target platform, there are tools available for that such as: JQuery Mobile and Sencha Touch [12].

### 4.5.2 Ionic

Another hugely successful framework is the Ionic Framework. It is also based on top of Apache Cordova. Cordova deals with the low level hardware hooks allowing Ionic to use various hardware features on a device, such as the accelerometer, gps and camera. Ionic was created by Drifty Co and is an open source framework released in 2013, it is based on AngularJS and is open source under the MIT license. Ionic deals mostly with the visual representation of the application, you develop the application once and you can then compile it to either iOS or Android and it keeps the native look and feel. There are many widgets available via Ionics library in order to achieve the same standard as native 5 apps. You would not be able to discern many Ionic apps against the native ones. They also include services for analytics and push notifications All you need to know in order to develop Ionic applications is CSS, HTML5, and JavaScript. Applications are than distributed to app stores such as Android and iOS. Ionic requires NPM (node package manager) in order to install various plugins. Ionic seems to be one of the most popular cross platform development frameworks for mobile at the moment.

## 4.6 Generated Applications

These applications are compiled natively, for that reason they achieve high performance and generate a native user interface. One of the frameworks used is Applause [14]. Applause is open source and is based on Xtext but not much development has been done on the framework [13]. Applause is explained in more detail down below.

### 4.6.1 Applause

Applause is based on model driven software development (MDSD). It includes tools to translate programming languages. It uses cross-compilers (XMLVN) [15] and transforms Android applications in Java. Applause is under EPL license which integrates with Eclipse and IDE support. There was very little information available on Applause, there is currently only one GitHub link [14].

## 4.7 Web Applications

Web applications are browser-based applications running in a browser using HTML5. WebHooks allow developers to access the hardware on a phone, this was unavailable before HTML5 [16], also it allows other features such as web storage, indexed database APIs, file APIs, web SQL Databases and Offline Web GeoLocations [16]. This makes web applications more mobile friendly and compatible and allows them to use the full range of phone features. They do not require installation or any upgrades as it contains a one to many relationship (one server, many clients) so any updates are done on the server side and all clients get updated, but the network is required at all times in order to access the application. It lacks the native look and feel of target platforms such as Android, iOS or Windows Phone, altough there are many tools out there trying to solve the problem by simulating a native look such as Xui, JQyeryMobile, Sencha Touch, JQTouch and WebApp.net. Some 6 frameworks developing web applications include AngularJS, Ruby on Rails, Django and Drupal. AngularJS is explained in more detail down below.

### 4.7.1 AngularJS

It was developed by Misko Hevery in 2009 at Brath Tech LLC. It is now an open source framework mainly used for developing single page applications

(SPA) it has become widely well known and is the top choice for many developers for creating dynamic html pages. In order to be able to program in AngularJS you have to know HTML, CSS and JavaScript. It is maintained by Google and the developer-community; it is under MIT license [17]. It uses data binding which means you can attach controllers to certain parts of the page as well as taking advantage of the MVC (Model, View, Controller) pattern, creating a loosely coupled design to separate the three components of the web application so that they all are independent to one another; one of them can be changed without impacting the others and you can swap and change components. If an application contains more than one page it can use Client side routing in order to dynamically switch content without refreshing the page [18]. The Batarang plugin was built by Google in 2012 to improve the debugging of web applications built using AngularJS. It is also used with another three popular technologies known collectively as the MEAN Stack (MongoDB, Express, AngularJS and NodeJS). MongoDB is cross platform oriented database, it uses a JavaScript/JSON style syntax; it is open source. Express is a server framework that is used for building single page web applications and is expandable via plugins. NodeJS is cross platform runtime environment for server side applications, it's open source. As we can see all the technologies used in the MEAN stack are open source suggesting the reason for its huge community and popularity.

## 4.8 User Interface User Experience

There are many different operating systems out there like Google's Android, Apple's iOS, Microsoft 's Windows Phone OS, Nokia 's Symbian, Blackberry's OS, Samsung's BADA, WebOS and HP. The devices these OS' run on all have many different screen sizes. Each platform has a different user interaction experience and two main platforms out there would be Android and iOS; both of them have a different user experience. It is emphasised to programmers that they should follow each platforms various rules and guidelines, especially with iOS [19]. The mobile user interface is very different to that of the desktop and some companies (like Microsoft) tried to merge the experience with the Windows Metro GUI (Graphical User Interface) which was released in Windows 8, they quickly realised this was a 7 mistake after the public backlash and so they rolled back to the classic look for Windows 10. Many users like the distinction between the desktop and the mobile interface. Native applications always provide the highest level of native user experience because it's compiled to native code for the platform that is being used on. As described in earlier sections there are frameworks that fully

achieve a native look for applications, a few of the examples would be Titanium and Applause. There are others which do not fully achieve native the native look, like PhoneGap and AngularJS but they use various libraries to achieve more of a native look, libraries such as Xui, JQyeryMobile, Sencha Touch, JQTouch and WebApp.net. According to results produced by paper [20] a sufficient level of native user interface can be obtained if an appropriate framework is picked to develop an application.

## 4.9 Conclusion

9. Conclusion "Write once run anywhere" [3] is a principal that was coined by Sun Microsystems when they were developing java, which is exactly the same as what all of these frameworks are trying to achieve. We still see heterogeneity within cross platform development as it is split into four groups as described earlier on. We see some categories gaining huge popularity and being adopted by the community very well, like web applications with HTML5 allowing developers to use the phones hardware. With popular web application frameworks like AngularJS maintained by Google and as part of a bigger bundle with the MEAN Stack and with the new release of JavaScript's ECMAScript standard 6 it will make it more usable, robust and will add extra capabilities. While some other frameworks such as Applause came into existence and didn't gain much interest or popularity and not much development is being done as agreed by the developers from the Applause development team referenced in following paper [13]. So what we see here is a lot of development in Cross-Platform frameworks with some of them aiming at big markets like Android and iOS and others trying to cover all the operating systems. The number of different types of mobile devices keeps increasing and with the rise of IoT (internet of things) more and more devices are getting connected to the Internet, which leads to an exponential curve, meaning more platforms and more cross platform development.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XMl, WSDL, JSON, JAXP.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

## 4.10 XML

Here's some nicely formatted XML:

```
<this>
  <looks lookswhat="good">
    Good
  </looks>
</this>
```

# Chapter 5

# System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

| Column 1 | Column 2 |
|---|---|
| Rows 2.1 | Row 2.2 |

Table 5.1: A table.

# Chapter 6

# System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportuni-ties in your approach or technologies used.

# Chapter 7

# Conclusion

About three pages.

- Briefly summarise your context and ob-jectives (a few lines).
- Highlight your findings from the evalua-tion section / chapter and any opportuni-ties identified.