# How Does a Typical Tutorial for Mobile Development Look Like?

Rebecca Tiarks
University of Hamburg
Department of Informatics
Hamburg, Germany
tiarks@informatik.uni-hamburg.de

Walid Maalej
University of Hamburg
Department of Informatics
Hamburg, Germany
maalej@informatik.uni-hamburg.de

## ABSTRACT

We report on an exploratory study, which aims at understanding how development tutorials are structured, what types of tutorials exist, and how official tutorials differ from tutorials written by development communities. We analyzed over 1.200 tutorials for mobile application development provided by six different sources for the three major platforms: Android, Apple iOS, and Windows Phone. We found that a typical tutorial contains around 2700 words distributed over 4 pages and including a list of instructions with 18 items. Overall, 70% of the tutorials contain source code examples and a similar fraction contain images. On average, one tutorial has 6 images. When analyzing the images, we found that the studied iOS community posted the largest number of images, 14 images per tutorial, on average, from which 74% are plain images, i.e., mainly screenshots without stencils, diagrams, or highlights. In contrast, 36% of the images included in the official tutorials by Apple were diagrams or images with stencils. Community sites seem to follow a similar structure to the official sites but include items and images which are rather underrepresented in the official tutorials.

From the analysis of the tutorials content by means of natural language processing combined with manual content analysis, we derived four categories for mobile development tutorials: infrastructure and design, application and services, distribution and maintenance, and development platform. Our categorization can help tutorial writers to better organize and evaluate the content of their tutorials and identify missing tutorials.

## Categories and Subject Descriptors

D.2.6 [**Software Engineering**]: Metrics – Process Metrics; I.7.1 [**Document and Text Processing**]: Document Capture – Document analyses

## General Terms

Human Factors, Documentation

## Keywords

Data mining, Software documentation, Knowledge sharing

## 1. INTRODUCTION

Developers spend time exploring new technologies or unfamiliar source code [11]. They have to decide which parts of these technologies or code are relevant to their current task [23], what kind of actions they want to take, and which information they need in their current context. Procedural knowledge or "how-to knowledge" is thereby an important type of knowledge, which describes how to perform a specific task. Procedural knowledge is very useful [13] but often exists in the minds of experienced developers or is documented informally. Sharing procedural knowledge can be achieved by persisting it in form of help documents, frequently asked questions (FAQs), or tutorials.

Tutorials aim at transferring knowledge and teach a process to the tutorial user. A general definition states that a *tutorial is a book, computer program, etc., that teaches someone how to do something by explaining each stage of a process*[1]. Vestadm [24] focus on program tutorials and define those as *documents that inform the user about the internal properties of a program, especially with the goal of using the program for construction of composition of another program.*

Tutorials usually describe a process as a set of instructions, examples, explanations, and images to teach the reader how to perform a specific task. Tutorials exist on a wide range of topics from a tutorial on how to use a specific tool such as Dropbox to tutorials on how to start working with a programming language like Objective-C. They are either provided on community sites and blogs [17], or as part of the official documentation of specific technologies, projects, and organizations.

Tutorials have several advantages compared to other types of documentation, as they are interactive and people learn better when they perform actions themselves. Readers typically see a tutorial as a "crash course" through which they can quickly get introduced to a new technology. For example, as mobile app development is getting more and more popular within a short period of time and the number of available technologies, Application Programming Interfaces (APIs), and development environments is increasing rapidly, tutorials are becoming more and more popular amongst developers for going with these trends.

Although there are several studies about software documentation in general, and its value for the knowledge seekers and knowledge providers (see Section 5), studies about development tutorials are rather rare. It remains unclear what is a "good" tutorial, how does a typical tutorial should look like, and what type of knowledge should be shared and

---

[1]http://www.merriam-webster.com/dictionary/tutorial

accessed where. As a first step for answering these questions and to reason about the quality and value of tutorials, we must first understand what knowledge tutorials contain, and how this knowledge is organized. We report on an exploratory study that aims at understanding what kinds of tutorials exist, how tutorials are structured, and how official tutorials differ from tutorials written by development communities.

The contribution of this paper is threefold. First, it gives empirical evidence on what a tutorial for mobile development looks like and what are its structural elements. Second, it gives insights on topics that are covered by tutorials and their popularities. Third, it identifies and discusses differences between official tutorials and tutorials written by development communities.

The reminder of the paper is organized as follows. Section 2 describes our research setting in including the questions, data, and methodology. Section 3 reports on the results of the structure analysis and Section 4 of the content analysis of mobile development tutorials. Section 5 surveys related work. Section 6 and 7 discuss the findings of our study and its limitations. Finally, Section 8 summarizes the paper and points to future directions.

## 2. RESEARCH SETTING

In this section we describe our research goal and the overall method we used to collect and analyze the data.

The term tutorial is only loosely defined and there is no standard guideline on how tutorials are supposed to be written. In general a tutorial teaches someone something new. There are different types of tutorials available on the Internet. Tutorials can be written documents, interactive programs, or screen recordings.

In this paper, we study written tutorials. We focus on two aspects of tutorials: (a) *the structural elements*, that is the set of instructions, examples, explanations, and images in a tutorial and (b) the aspect of teaching the reader *a specific knowledge* to solve a specific *task*. To the best of our knowledge there is no empirical study that analyzes the structure and content of tutorials.

### 2.1 Research Questions

We study the **structure** and **content** of development tutorials. Tutorial structure describes *how* information is presented in the tutorials and how are typical structural elements used. We answer the following specific questions:

- **Length:** How long are tutorials?

- **Lists and sections:** How are structural elements such as sections and lists used in tutorials?

- **Source code:** How much source code is included in tutorials?

- **Images:** What images are included in tutorials?

- **Links:** How many links are included in tutorials?

- **Evolution:** Are there changes in the tutorial structure between different revisions of the system documented by the tutorial?

In order to analyze the content we identify topics that are covered by tutorials. We also analyze the frequencies of the identified topics.
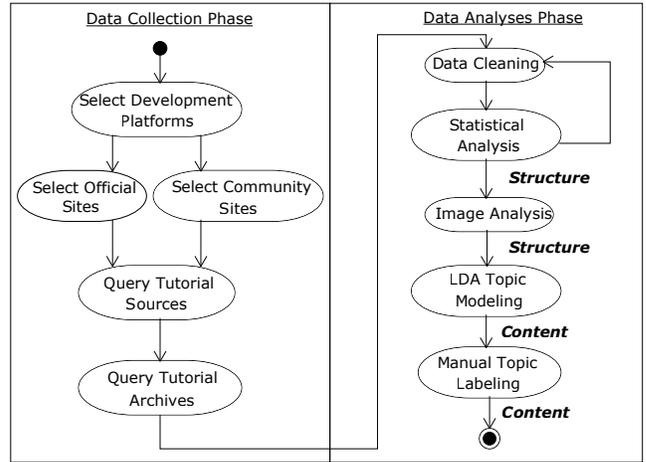


**Figure 1: Research method.**

- **Topics:** Which topics are covered by development tutorials?

- **Topic popularity:** How popular are the topics across different types of tutorials?

When answering these questions we distinguish between **official tutorials**, that is those being provided by the official pages of the technology or platform being documented and the **community tutorials** which are written by famous developers and broadly used in the community.

### 2.2 Research Method

Our research method is divided into two phases: a data collection and a data analysis phase. The whole process is depicted in Figure 1.

#### 2.2.1 Data Collection

In the data collection phase we selected the official and the community sites, from which we collected our data. Further, we chose an Internet archive[2] from which we collected previous revisions of the tutorials. We used jsoup[3] a Java library to extract data from HTML documents. We developed a crawler that traverses the document tree of a tutorial source and downloads the contents in form of HTML sites and images.

To qualify for our study the tutorials had to fulfill the following requirements:

- The tutorials have to be related to the field of mobile application development.

- The content is publicly available.

- The content describes a process and contains some form of instructions.

#### 2.2.2 Data Analysis

The data analysis phase consisted of four steps. First we cleaned the downloaded data. Our crawler has no selection method implemented and downloaded all documents available from a given website. Therefore reference documents

---

[2]http://archive.org/web/
[3]http://jsoup.org

and release notes were also included in the downloaded data. We removed these documents by using regular expressions.

To extract the contents of the HTML documents we traversed the elements of the dom tree, created by jsoup for each tutorial and selected the relevant tags. For each analysis we had to find out which tags are used in the different tutorials to markup the specific elements. The tags differed from source to source. In order to analyze the *structure* of the tutorials we applied descriptive statistics. This analysis also helped to identify further documents that were not tutorials but API reference documents, release notes, or other documents. We applied descriptive statistics to the current tutorials and to those collected from the Internet archive.

To answer the question what types of *images* are included in mobile development tutorials we first inspected the images manually and performed an automatic image analysis afterwards by using the Java implementation of OpenCV[4].

To analyze the *content* we used the Latent Dirichlet Allocation [3]. This topic modeling technique is based on the assumption that each document consists of a number of topics and that there is a latent structure in the texts that are analyzed. A topic is represented as a set of words and each word has a certain probability to be correlated with a topic.

Before we could analyze the tutorial content we had to perform several preprocessing steps, which we will describe in the following. First, we extracted the plain text from the tutorials. We also extracted the text of html elements like paragraphs (<p>), headlines (<h1>, <h2>,..) lists (<ol>, <ul>), and list items (<li>). After that, we removed stopwords with a predefined list of English stopwords. For the next optimization we used a stemmer, namely the Porter Stemmer [20]. The algorithm removes word inflexions and thus normalizes terms.

## 2.3 Research Data

We selected six large sources of tutorials for mobile applications. We compare official sites of Apple[5], Android[6] and Windows[7] to independent community sites, namely iOS tutorials from Ray Wenderlich[8], Android tutorials by Lars Vogel[9] and Windows Phone tutorials by Kirupa Chinnathambi[10]. The Ray Wenderlich site is a blog focused on developing programming tutorials and the team behind the site consists of over 100 people from which 50 people are related to tutorial writing and reviewing. The team from the vogella GmbH who are responsive for the Android community website consists of six people. Nevertheless the Android tutorials are all written by Lars Vogel himself. This also applies to the Windows Phone tutorials which are written by Kirupa himself but the team consists of over 30 members which are responsible for the forum.

Table 1 shows the summary of our data. Our first mobile development platform is Android. Android is based on the Linux kernel and the source code released by google is under the Apache License. This allows modifications and distributions of the software. The first mobile device platform of Android was released in 2007. From the official Android

---

[4]http://opencv.org
[5]https://developer.apple.com/library/ios/navigation/
[6]http://developer.android.com/develop/index.html
[7]http://dev.windowsphone.com/en-us/
[8]http://www.raywenderlich.com/tutorials
[9]http://www.vogella.com/tutorials/android.html
[10]http://www.kirupa.com/me/index.htm

**Table 1: Overview of the dataset used in the study.**

| Tutorial | # tutorials | size (#words) | avg. size (#words) |
|---|---|---|---|
| Android community | 48 | 108.891 | 2.269 |
| Android official | 91 | 530.528 | 5.830 |
| Apple community | 469 | 1.368.306 | 2.917 |
| Apple official | 557 | 2.013.937 | 3.616 |
| Windows community | 87 | 62.291 | 716 |
| Windows official | 22 | 19.139 | 870 |

site we extracted 48 tutorials and 91 from the community site. Apple introduced its iPhone also in 2007 with the first version of its operating system iOS. For Apple there are the most tutorials available. We extracted 469 tutorials from the community site and 557 tutorials from the official site. The newest mobile development platform is Windows Phone developed by Microsoft. It was released in 2010 and thereby offers the smallest amount of tutorials available, for both the community and the official site. The community site consists of 87 tutorials and the official site only contains 22 tutorials.

## 3. TUTORIAL STRUCTURE

To study the structure we applied several measures on the tutorials, including *source code, links* and *images.* In the following section we describe our results.

### 3.1 Length

The median tutorial length is 3038 words (3231 words average). Figure 2 shows the distribution of tutorial length. In all datasets the distribution is positively skewed since there are some tutorials that comprise over 10000 words. The longest tutorial is from the official Apple site containing 47141 words. The tutorial describes how to design human interfaces for iOS. The smallest tutorial containing only 54 words is also from the official Apple site. It provides links to iOS resources and briefly describes them.

Tutorials for Windows Phone are shorter than the tutorials for the other platforms. The official Windows tutorials only contain 870 words, on average, (648 words median) and the community tutorials only 716 words (683 median). Nevertheless the Windows tutorials are more equally distributed
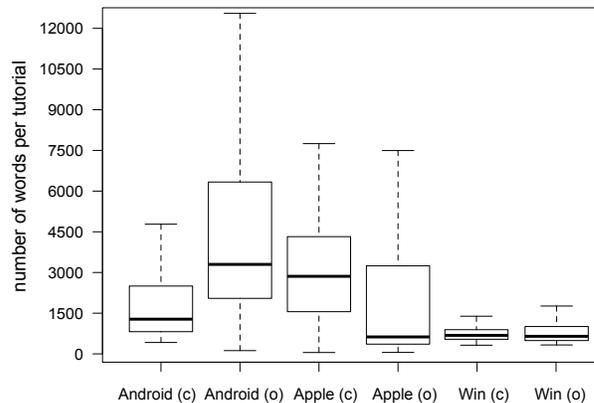
**Figure 2: Length of mobile development tutorials.**

in length as the smallest tutorials contain around 300 words and over 75% of the tutorials contain more than 500 words (for both Windows community and official). The longest tutorials, on average, are from the official Android site ( 3298 words median). Over 75% of the tutorials are longer than 2000 words.

There is a significant difference between the tutorial lengths across the studied platforms. Android tutorials seem to be the longest, followed by Apple, and Windows. Within the same platform, a community tutorial seems to follow the length of an official tutorial.

## 3.2  Lists and Sections

We assume that tutorials typically contain instructions. Instructions are often represented as lists (i.e. unordered/bullet lists and ordered/numbered lists). Therefore we analyzed the number of lists contained in our tutorials. The results are depicted on Figure 3. Most unordered lists are contained in the official Android tutorials (9 lists median) and most ordered lists are contained in the official Windows tutorials (2 lists median).

Regarding the contents of a list the lists of the official Android tutorials contain most list items as Table 2 shows. The smallest lists, on average, are contained in the Windows community tutorials (5 items per list mean). Surprisingly the items from the Windows community contain the largest amount of words (25 words mean) and the Android official items contain the smallest amount of words (12 words mean).
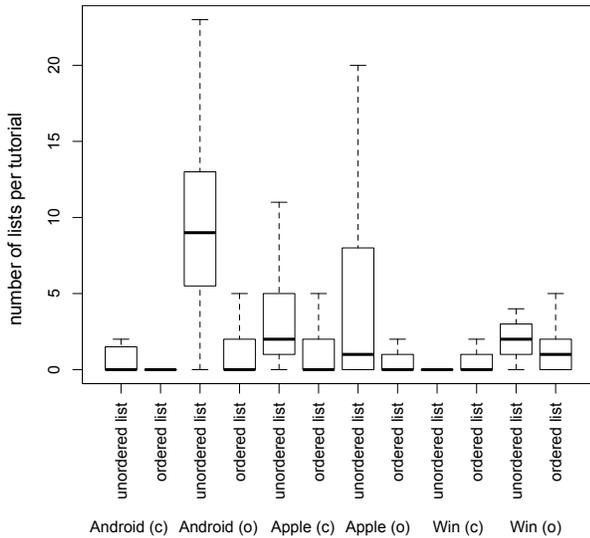
**Table 2: Average number of items per list and number of words per list item in the tutorials.**

| Tutorial | Avg. items per list | #words per item |
|---|---|---|
| Android community | 11 | 10 |
| Android official | 34 | 12 |
| Apple community | 15 | 20 |
| Apple official | 31 | 21 |
| Windows community | 5 | 25 |
| Windows official | 10 | 15 |

tutorials seem to have a rather large number of sections (18 to 22), Apple community and official have a rather moderate number (2 to 7), and Windows rather small number of sections (1 to 3).

The results show that the official tutorials typically include more structural elements such as sections and bulleted lists than the community tutorials. The Apple official tutorials have a large variance in the number of structural elements. It seems that there are two different types of Apple official tutorials: those with only a few structural elements and those with a more detailed structure.
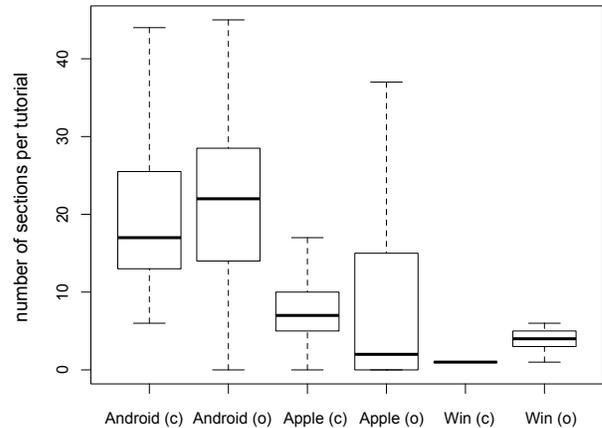


**Figure 4: Number of sections contained in mobile development tutorials.**

## 3.3  Source Code

A total of 874 tutorials contained source code (70%). We compared the official sites to the community sites and the only difference can be found for Windows where 68% of the official tutorials contain source code and only 36% of the community tutorials.

To measure the ratio of source code compared to text we compared the number of characters of source code to the characters extracted from the text. Table 3 shows the ratio of source code for the analyzed tutorials. The fourth column shows the ratio between text in source code only for tutorials that contain source code. The fifth column summarizes the ratio of source code measured over all tutorials. The official sites from apple contain the largest source code samples with a median of 10.536 characters (35.496 characters on average).



**Figure 3: Number of lists contained in mobile development tutorials.**

We also analyzed the number of sections contained in a tutorial. Figure 4 shows the results. The various tutorials have a varying number of sections. Most sections are contained in the official Android tutorials (22 sections median). We found the smallest number of sections per tutorial for the Windows community (1 section median). Those tutorials only consist of one section. Interestingly, the community and official tutorials are similarly structured for each platform. On average, while Android community and official

**Table 3: Source code in mobile development tutorials.**

| Tutorial | # Tutorials with source code | Avg. source code length (chars) | Source code/text ratio | Source code/text ratio (total) |
|---|---|---|---|---|
| Android Comm. | 44 (92%) | 8.309 | 54,11% | 44,04% |
| Android Official | 77 (85%) | 12.030 | 22,81% | 21,14% |
| Apple Comm. | 324 (69%) | 16.626 | 28,29% | 25,48% |
| Apple Official | 383 (69%) | 35.496 | 56,98% | 51,51% |
| Windows Comm. | 31 (36%) | 915 | 17,26% | 7,41% |
| Windows Official | 15 (68%) | 3.070 | 35,54% | 27,41% |

**Table 4: Links in mobile development tutorials.**

| | Tutorials with links | Tutorials with links to API |
|---|---|---|
| Android Community | 48 (100%) | 3 (0,06%) |
| Android Official | 91 (100%) | 76 (84%) |
| Apple Community | 469 (100%) | 253 (54%) |
| Apple Official | 557 (100%) | 180 (32%) |
| Windows Community | 87 (100%) | 0 (0%) |
| Windows Official | 22 (100%) | 22 (100%) |

Overall, the results show that source code is intrusive in development tutorials for both official and community tutorials, except for Windows community, which also has the smallest source code samples with a median of 836 characters (915 characters on average). The median length of source code for the official Windows site is 2.069 characters. The median lengths of the remaining three sites are at around 7.000 characters (6.486 Android community, 7.281 Android official and 7.170 Apple community).

## 3.4 Links

We analyzed the number of links contained in tutorials. The results are summarized in Table 4. We observed that all 1.274 tutorials contain links. To further investigate the type of links, we automatically extracted the links and used regular expressions to find links that direct to API reference documentation for each development platform. The community sites overall contain less API links (18%) compared to the official sites (46%). Nevertheless the overall average does not account for the fact that this relation only applies to Android and Windows and not for Apple. To our surprise the official sites from Apple contained only 32% API links whereas the community sites for iOS contained 54%. The community sites from Android and Windows contained no or nearly no API links.

Figure 5 shows the distribution of links and API links in the tutorials. The official sites from Android contain the largest number of links (55 links median) and API links (85 links median). Over 75% of the tutorials contain more than

59 links. The smallest number of links is contained in the Window community sites (9 links median).

Overall, official tutorials include significantly more links to APIs than community tutorials, except for Apple. In this case, the community tutorial are complementary and have more API links, which seem to be underrepresented in the Apple official tutorials compared to Android and Windows.
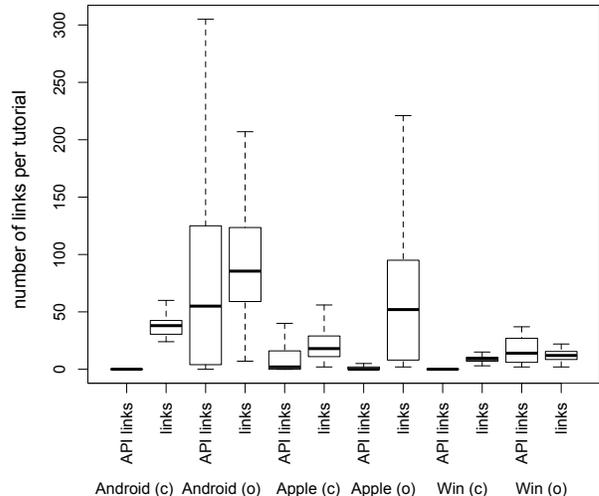


**Figure 5: Number of links in mobile development tutorials.**

## 3.5 Images

We analyzed the occurrence of images in our tutorial dataset. Table 5 shows the results. We found that 885 tutorials contain images (69%). Most images per tutorial are contained in the Apple community tutorials (median is 10).

In order to find out what types of images are contained in the tutorials we manually investigated all 11.266 images. We distinguished between plain images (mainly screenshots), images with stencils or highlights, and diagrams. Images that did not belong to these categories were not taken into account, e.g., icons and small images of buttons (for navigation such as *previous* and *next)* or images that are not related to the tutorial content (e.g. smileys). Table 6 summarizes the results from our analysis.

We also performed an additional automatic image analysis to find stencils in tutorial images. We used OpenCV, an open source library for image processing. We employed the edge detection function and some other methods provided

**Table 5: Images in mobile development tutorials.**

| | Tutorials with images | Avg. # images in a tutorial | Std. dev. |
|---|---|---|---|
| Android Community | 32 (100%) | 3 | 6 |
| Android Official | 56 (64%) | 6 | 16 |
| Apple Community | 458 (98%) | 14 | 15 |
| Apple Official | 205 (36%) | 4 | 15 |
| Windows Community | 87 (100%) | 4 | 4 |
| Windows Official | 20 (91%) | 2 | 2 |

**Table 6: Image types in mobile development tutorials (manual analysis).**

| | # Images without stencils | # Images with stencils | #Diagrams | #Skipped |
|---|---|---|---|---|
| Android Community | 110 (69%) | 25 (15,6%) | 20 (12,5%) | 5 (3,2%) |
| Android Official | 350 (58%) | 87 (14,5%) | 109 (18,2%) | 53 (8,8%) |
| Apple Community | 5.105 (74%) | 1.200 (17,3%) | 299 (4,3%) | 330 (4,8%) |
| Apple Official | 1.530 (56%) | 281 (10,2%) | 714 (26,0%) | 223 (8,1%) |
| Windows Community | 346 (46%) | 98 (12,9%) | 31 (4,1%) | 292 (37,3%) |
| Windows Official | 37 (54%) | 11 (16,2%) | 2 (2,9%) | 18 (26,5%) |

**Table 7: Image types in mobile development tutorials (automated analysis).**

| | #Images w. red rect. stencils | # Images analyzed |
|---|---|---|
| Android Community | 13 (9,2%) | 141 |
| Android Official | 13 (2,5%) | 515 |
| Apple Community | 95 (1,4%) | 6.671 |
| Apple Official | 192 (8,4%) | 2.283 |
| Windows Community | 10 (2,0%) | 488 |
| Windows Official | 5 (11,9%) | 42 |

**Table 8: Tutorials from 2013 compared to current tutorials (from 2014).**

| Tutorial | # tutorials | size (#words) | avg. size (#words) |
|---|---|---|---|
| Android community | 48 (0%) | 100.304 (-8%) | 2.089 (-180 words) |
| Android official | 53 (-41%) | 225.532 (-57%) | 4.255 (-1575 words) |
| Apple community | 172 (-63%) | 470.987 (-66%) | 3.708 (+791 words) |
| Apple official | N/A | N/A | N/A |
| Windows community | 87 (0%) | 61.752 (-1%) | 709 (-7 words) |
| Windows official | 16 (-27%) | 14.070 (-26%) | 879 (+9 words) |

by OpenCV to extract red rectangles from the images. The analysis is limited as some images were too dark to distinguish the stencils from the rest of the image and some images could not be analyzed at all. Our analysis is also limited to red rectangles, which means that we missed all stencils that are shaped differently (circles, ellipses) or had other colors.

The results in Table 7 show a similar tendency for three of the six tutorial sources, namely Windows official, Android community, and Apple official. Regarding the remaining three sources the results were different from the results produced by the manual analysis. This is due to the fact, that the images were either too dark or contained other stencils than red rectangles.

### 3.6 Tutorial Evolution

We studied the evolution of tutorials for five of the six tutorial sites in order to explore if there are changes in the length of tutorials between two different revisions of tutorials sources. The results are shown in Table 8. The percentage indicates the increase or decrease compared version of the current tutorials retrieved in 2014. A negative value means that the older revision is smaller than the current version.

Comparing the current number of tutorials to the number of tutorials ten month ago the number of tutorials increased for all analyzed sources. The Apple community tutorials increased most (63%). Regarding the length of the tutorials the total amount of words also increased for all tutorial sources. Surprisingly, the amount of words more than doubled for the official Android site (57%) and the Apple community site (66%).

Similarly, the number of images increased proportionally and also more than doubled for the official Android site (69%) and the Apple community site (59%).

The ratio of tutorials that contain source code increased or remained equal for four of the five analyzed tutorial sources. Surprisingly, the ratio of tutorials that contain source code from the Apple community site decreased from 80% to 69%.

This means that the number of tutorials doubled within ten month but the number of tutorials that contain source code did not increase proportionally.

Finally, we analyzed the average number of words for each tutorial. We found that the average number of words decreased significantly for the Apple community tutorials, which means that the current tutorials contain 791 less words, on average, than the revision from 2013.

### 4. TUTORIAL CONTENT

This section summarizes the results of our LDA analyses [3] and manual topic labeling. We identified 17 topics covered by mobile development tutorials. We also grouped the topics into four more general categories.

### 4.1 Topics in Tutorials

We analyzed the content of the mobile development tutorials to find out which topics are covered in them.

In order to find the parameters that produce the best results, we performed several runs of the LDA on our dataset with different numbers of topic, iterations, and thresholds. We also adapted our stopwords after the first iterations, as we found that words like *about* appeared very often and were not included in the stopword list.

The more documents are available the higher is the number of topics that can be identified. As the number of documents varied for the tutorial sources we obtained the best results with a varying number of topics. For the Android and Apple tutorials we chose 50 topics, for the Windows tutorials we reduced the number to 20 topics. Overall we obtained a list of 240 topics.

We then manually inspected the top 20 most influential words we obtained from LDA for each topic and added labels that describe the meaning of the topic. The last step was grouping similar topics together across all datasets.

The identified topics and their occurrences are shown in Table 9. The occurrences are calculated by calculating the occurrence of each topic within the tutorials. This number was extracted from the document-topic matrix obtained from the LDA analysis. One tutorial can contain several topics and the contribution to a topic is measured by the number of words that belong to the topic. We chose the most predominant topics per tutorial. A topic is predominant if at least 10% of words from the tutorial belong to the topic. We derived this threshold from the observation that over 90% of all topics contained one predominant topic.

We found that the most frequent topic is "architecture and frameworks". This topic was mentioned in tutorials that are specific for a single mobile development platform and are not applicable to all platforms, e.g., ARC from Apple (automatic reference counting) or Windows predefined behaviors.

Equally predominant in 30% of all tutorials is the topic GUI, layout, and graphics. We think that this because all most mobile applications provide a user interface and therefore the tutorials describe topics related to layout and the appearance. The topic app store and deployment is predominant in one fourth of all tutorials. For the official sites of Apple even one third of all tutorials contain the deployment topic. In over 20% of all tutorials the topic development environments and frameworks is predominant which means that also issues about development tools, their configuration, and usage are common in the tutorials.

Interestingly, nearly 12% of all tutorials cover the topic games. We found the largest number of tutorials covering games in the Apple community sites (28%). These tutorials often use games as examples for domain specific applications. The topic problems and solutions (e.g. typical errors and how they should be handled) is only covered in less than 3% of all tutorials. We think that this kind of topics is rather addressed in other forms of documentation such as frequently asked questions (FAQs).

## 4.2 Categories of Topics

We grouped our topics into more abstract categories to further investigate the content of the tutorials.

1. Infrastructure and design: The topics in this category are related to *architecture and frameworks* of mobile development. The category also compromises topics about *data, storage, access, permissions* and *connectivity.*

2. Application and services: This category describes topics that deal with application and services for mobile development like *GUI, layout and graphics, games, media, location services, animations* or *event and gesture handling.*

3. Distribution and maintenance: These topics describe information on *deployment* issues like *licensing,* or *hardware, screens* and different devices. *Testing and debugging* belong to this category as well as *problems and solutions* because testing and error solution are corrective tasks.

**Table 9: List of identified topics across all datasets.**

| Topic description | Occ. offic. | Occ. comm. | Influential words |
|---|---|---|---|
| Architecture & frameworks | 41,8% | 55,1% | arc, components, behavior, manifest, class |
| App store, deployment, and licensing | 29,9% | 22,5% | app store, google, play, developer, product, billing, purchase |
| GUI, layout, and graphics | 20,8% | 38,4% | drawing, color, image, graphic, view, controller, |
| Development environments | 14,5% | 30,3% | eclipse, development, make, build |
| Application lifecycle | 13,4% | 4,1% | lifecycle, background, oncreate, activity |
| Data storage and I/O | 12,8% | 5,0% | storage, database, backup, save, file |
| Multimedia | 11,6% | 3,9% | camera, image, audio, sound, volume, player, stream |
| Events and gestures | 9,3% | 4,3% | event, touch, gesture, finger, position |
| Security, access, and permission rights | 6,4% | 3,0% | access, permission, rights, control |
| Testing and debugging | 6,0% | 5,8% | test, build, run, debug, log, trace |
| Games | 4,6% | 18,5% | game, action, sprite, player, score |
| Hardware, screens, and devices | 3,3% | 1,3% | screen, imageview, memory, resolution |
| Connectivity | 3,0% | 2,2% | bluetooth, network connection, wifi, ip |
| Animations | 0,3% | 4,0% | animation, texture, move, video, frame |
| Location services | 0,2% | 3,0% | location, map, coordinate, place |
| Community | 0,0% | 18,9% | review, reader, workshop, discuss |
| Problems, and solutions | 0,0% | 5,8% | problem, solution, question |

4. Development environments: This category includes the *development environments* topic.

Table 10 shows that tutorials often cover *infrastructure and design* topics such as architecture and frameworks. The category *application and services* is the most popular category in more than half of all tutorials for both community and official tutorials. We think that this is reasonable since developing applications is the major goal for users of mobile development tutorials. Interestingly, these tutorials also cover topics like *distribution and maintenance.*

We also found two significant differences in the topic popularity between official and community tutorials. The category *application and services* is more popular in community tutorials (72%) than in official tutorials (47%). This also applies to the category *development environments.*

**Table 10: Topic categories and their occurrence.**

| Category | Occ. offic. | Occ. comm. | Topics |
|---|---|---|---|
| Infrastructure and design | 78% | 69% | Architecture and frameworks, application lifecycle, data storage and I/O, security, access and permissions, connectivity |
| Application and services | 47% | 72% | GUI, layout and graphics, games, multimedia, location, animations, events, and gestures |
| Distribution and maintenance | 39% | 35% | App store, deployment and licensing, problems and solutions, testing and debugging, hardware screens, and devices |
| Development environments | 14% | 30% | Development environments |

## 5. RELATED WORK

In this section we discuss related studies of software documentation and tutorials. Software documentation captures and externalizes knowledge about the software. There are several studies that tried to categorize the types of this knowledge. Mylopulus [14] et al. studied different knowledge representation techniques from Artificial Intelligence and how they can be applied to software engineering. The knowledge types they presented were derived from their experience. They identified the following types: domain knowledge, requirements knowledge, design knowledge, quality factors, design rationale, and historical knowledge. Some of our topic categories are related to their knowledge types like infrastructure and design. The kinds of knowledge that may benefit from explicit capture at the requirements and design stages is described by Herbsleb and Kuwana [7]. To find these kinds of knowledge the authors classified questions asked in design meetings.

There are also a few studies that discussed the design, usability, and quality of documentation, especially API documentation. A study by Nykasa et al. [16] assessed the documentation needs for a domain specific API, using surveys and interviews. One of their results is the importance of an overview section in API documentation. A lab study with eight participants by Jeong et al. [8] assessed the documentation of a service-oriented architecture. The authors proposed 18 guidelines to improve documentation quality. Another study by Robillard and DeLine addressed the obstacles faced by developers when trying to learn new APIs [21, 22]. Dagenais and Robillard [5] reported on a study of how open-source contributors make decisions about the development and maintenance of API documentation by using surveys and interviews. A more recent study by Maalej and Robillard developed a taxonomy of knowledge types based on grounded methods and independent empirical validation [13]. The authors identified a knowledge type "non-information" and found that this type is very common

in the reference documentation of Java and .Net.

There is a large research field that applies data mining techniques to text created in development projects such as commits, work descriptions, and other documentation. To the best of our knowledge there is no empirical studies on the content and structure of online tutorials. Maalej and Happel [12] analyzed informal artifacts that summarize what developers have done in a work session. The personal notes and commit messages were analyzed by using natural language processing techniques.

Considerable effort was spent on the analysis of social media such as mailing lists and social networks [26]. Authors analyzed the explicit or latent structure of social networks as well as the individual or group behavior of social networks. Bird et al. studied the similarities of social networks created from developer email communication to development teams [2]. Maalej and Pagano [17] studied how developers and development communities use blogs. They found that open source communities intensively use blogs with one new entry about every 8 hours. They also used LDA topic modeling to derive a categorization of topics covered by blog posts and analyzed their popularity. Another study by Maalej and Pagano [18] analyzed over a million reviews from the Apple App Store and found that user reviews contain multiple topics, including documentation of the app features.

Only a few papers on development tutorials have been published. These often focus on the creation of programming tutorials or on tools that support the creation of documentation. Some approaches deal with programing tutorials but they are only limited to source code. The early version of programming tutorials is called literate programming and was proposed by Knuth [10]. Another variant of programming tutorials that is related to literate programming is called elucidative programing and was proposed by Nørmark [15].

Most research on tutorials has focussed on tutorials for users. In the field of documentation authoring systems, tools have been proposed to create static documentation (such as the Macromedia RoboHelp[11]). Bergmann et al. [1] proposed *DocWizards*, a system that is similar to RoboHelp and that builds upon scaffolding. The user is stepped through a script representation of a procedure by highlighting portions of the text and elements of the user interface. *DocWizards* infer general instructions from multiple demonstrations.

A number of systems use recording techniques to create tutorials. Common screen capturing tools such as Camtasia[12] record the interactions with application interface and produce tutorial documentation. Kelleher et al. [9] described systems that assist new users in learning a procedure. They use subtle guidance, so called scaffolding, to direct the user's attention to the relevant areas within an interface. A less intrusive form of scaffolding directs users thorough the process but leave them the freedom to deviate at will [25].

Finally, the studies by Harrison [6] and by Palmiter and Elkerton [19] evaluated different presentations of how-to knowledge. Harrison found that users using animated or illustrated text-based tutorials learned more quickly. Another study by Carrol et al. [4] found that users learned to use a program more quickly, when using a specially created train-

---

[11]http://www.adobe.com/de/products/robohelp.html
[12]http://www.techsmith.de/camtasia.html

ing version of a word processing package.

# 6. DISCUSSION

In this section we discuss the findings. We focused our analyses on three aspects: tutorial *structure*, *content*, and *evolution*. Regarding the structure, we first analyzed the length of the tutorials for the three studied platforms Apple, Android, and Windows. Overall, tutorials seem to be of a medium length including 4 pages of text on average. This is longer than informal documents such as blogs or forum posts [17] and perhaps shorter than more formal technical documentation. The Apple tutorials (both official and community) are the longest, followed by Android and then Windows tutorials. One possible reason is that Apple is the oldest platform and knowledge is accumulated in the tutorials over time.

We found that all tutorials include unordered and ordered lists. These results confirm our assumption that tutorials use *steps and instructions* as a form of information presentation. Lists contained in official tutorials are longer (that is with more items) compared to lists in community tutorials. This may lead to the assumption that the instructions in official tutorials are more detailed compared to community tutorials. A closer look to the number of words per list item reveals that the list items from the community tutorials include more words. In other words, official and community tutorials seem to include a similar amount of information, while the official tutorials are more structured and follow a fine grained step-by-step instructions.

The finding that over 70% of all tutorials contain source code shows that source code examples are an important type of information when describing how-to knowledge in tutorials. The same applies for links as all tutorials contained links. Surprisingly, there is no correlation between the number of API links and the tutorial sources. We expected a higher amount of API links in the official tutorials but this is only valid for the official Android and Windows tutorials. The opposite applies for the Apple tutorials. Official iOS tutorials include a relatively small amount of API links. Overall, it seems that there is no common style and structure for tutorials across the studied platforms. Community sites seem to follow a similar structure to the official sites but include items and images which are rather underrepresented in the official tutorials. One potential reason is that community tutorials try to provide complementary information to official tutorials.

Comparing official tutorials with community tutorials revealed a few differences. The number of links, for example, was higher for all official tutorials. On the other hand the number of tutorials that contain images was higher for the community tutorials. However, this does not make any claims on the quality and usefulness of the images. Our manual image analysis revealed that all tutorial sources used roughly the same amount of images with highlights or stencils. Regarding the diagrams, the official Apple tutorials contained over 25% of diagrams. We think that these tutorials more often provide an overview of the architecture and frameworks by using diagrams.

Regarding the evolution of tutorials we found that tutorials evolve, possibly with the evolution of the system or platform being described. Our results show that the average size (in terms of words) and the number of tutorials that contain source code varies over time which may indicate that tutorials are adapted to the needs of the tutorial users. The length of the Apple community tutorials decreased by 791 words, on average, which may indicate that tutorial users prefer shorter tutorials. The same applies for the number of tutorials that contain source code. For the Apple community site the ratio decreased from 80% to 69%. The reason behind this may be that there is no need for a large number of source code examples.

We found that tutorials for mobile development often cover topics about infrastructure and design as well as specific services and applications such as games. The tutorials provide more than a summary of classes and interfaces. They also explain the rationale behind them and often guide the reader through an example how to use these interfaces in an instructional manner. We think that this is the main difference between tutorials and other types of documentation such as API reference documentation [13].

# 7. LIMITATIONS

Our study was not designed to be generalizable nor representative for all software development tutorials but we think that some of the results for mobile development can be applied to other domains. We aimed at an exploratory study to shed light on the structure and content of tutorials. The six selected tutorials sources were chosen to meet our requirements and cannot be seen as representative sample of mobile development.

We think that our results have an appropriate level of validity for the following reasons. First, our datasets included tutorials for three major mobile development platforms. Second, we performed several data cleaning steps to remove outliers and hazard factors. Finally, we used different analysis methods to confirm our results (descriptive statistics and topic analysis).

Despite these measures, there are some limitations that should be considered when interpreting the results. The extraction of relevant information from the tutorials involved defining the HTML tags from which the text should be collected. Since all tutorial sources have their own HTML structure and there are no standard tags for elements like source code our data extraction may have missed some content. We tried to reduce this threat by carefully inspecting the HTML sources and randomly checking the results of the HTML crawler.

Another limitation is caused by the fact that two of the three community sites are created by single persons. However we think that this does not bias our results as both the Windows and the Android sites provide tutorials in the same amount as the official sites. Moreover, these community tutorials are very popular within the respective community and dozens of people are involved in maintaining the tutorial sites. From our manual inspection we could not find any difference regarding the structure of tutorials written by a single person or tutorials written by more than one person.

Our automatic image analysis is rather experimental and the rate of false negative results might be high. There are also limitations in the type and color of shape it can detect. Nevertheless we think that the results are worth reporting as they can be combined with the results of the manual analysis.

Finally, part of our results relies on manual analysis such as the manual image analysis or the labeling and categorization of topics. These results are subject to the experimenter

bias. To overcome this limitation both authors reviewed the categories independently.

## 8. CONCLUSION

What are typical elements of development tutorials and what topic do they cover? In this exploratory study we found that a tutorial contain, on average, 2.700 words over 4 pages. Typically, tutorials include itemed instructions that are presented in form of ordered and unordered lists as well as sections. However the number of items and sections varies across the studied platforms. Official tutorials seem to be more structured than community tutorials. Source code examples and images are essential parts and are included in around 70% of all 1.274 studied tutorials. Regarding the topics, *architecture, frameworks*, *GUI, layout, and graphic*s are predominant in the tutorials for mobile development.

Our results represent a starting point for characterizing the quality of tutorials and the efficiency of creating and using them. We believe that tutorials as valuable documentation type should be further explored to understand their role in the development process. Next steps include studying the usage of tutorials in order to evaluate which information (e.g. topics) is actually used from tutorials and which information is missing. In addition we plan to analyze tutorials from other fields of software development to evaluate the generalizability of our results to other fields. We also think that the evolution of tutorials is worth further investigation since we only analyzed how the size and length evolved over time and only for two revisions that are ten month apart. Deeper investigation could reveal trends in the evolution of tutorials and give insights on their usage and how they are adapted to the needs of the tutorial readers.

Our study motivates additional research in the field of automated classification of knowledge in tutorials. It provides first insights on knowledge types that are specific to tutorials and which may not be included in other documentation.

## 9. ACKNOWLEDGEMENT

## 10. REFERENCES

[1] L. Bergman, V. Castelli, T. Lau, and D. Oblinger. Docwizards: a system for authoring follow-me documentation wizards. In *In Proc. ACM UIST*, pages 191–200. ACM Press, 2005.

[2] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *MSR'06*, MSR '06, pages 137–143, New York, NY, USA, 2006. ACM.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[4] J. M. Carroll and M. B. Rosson. Paradox of the active user. In J. M. Carroll, editor, *Interfacing thought: cognitive aspects of human-computer interaction*, chapter Paradox of the active user, pages 80–111. MIT Press, Cambridge, MA, USA, 1987.

[5] B. Dagenais and M. P. Robillard. Creating and evolving developer documentation: Understanding the decisions of open source contributors. In *SIGSOFT'10*, pages 127–136, November 2010.

[6] S. M. Harrison. A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. In *SIGCHI '95*, CHI '95, pages 82–89, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[7] J. D. Herbsleb and E. Kuwana. Preserving knowledge in design projects: what designers need to know. In *Proceedings of the Joint INTERACT '93 and CHI '93 Conferences on Human Factors in Computing Systems*, pages 7–14, 1993.

[8] S. Y. Jeong, Y. Xie, J. Beaton, B. A. Myers, J. Stylos, R. Ehret, J. Karstens, A. Efeoglu, and D. K. Busse. Improving documentation for eSOA APIs through user studies. In *Proc. 2nd Int'l Symp. on End-User Development*, volume 5435 of *LNCS*, pages 86–105. Springer, 2009.

[9] C. Kelleher and R. Pausch. Stencils-based tutorials: design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 541–550, New York, NY, USA, 2005. ACM.

[10] D. E. Knuth. Literate programming. *Comput. J.*, 27(2):97–111, 1984.

[11] A. J. Ko, R. DeLine, and G. Venolia. Information needs in collocated software development teams. In *ICSE '07*. IEEE Computer Society, 2007.

[12] W. Maalej and H.-J. Happel. Can development work describe itself? In *MSR '10*. IEEE, 2010.

[13] W. Maalej and M. P. Robillard. Patterns of knowledge in api reference documentation. *IEEE Transactions on Software Engineering*, 39(9):1264–1282, 2013.

[14] J. Mylopoulos, A. Borgida, and E. Yu. Representing software engineering knowledge. *Automated Software Engineering*, 4(3):291–317, 1997.

[15] K. Nørmark. Elucidative programming. *Nordic J. of Computing*, 7(2):87–105, June 2000.

[16] J. Nykaza, R. Messinger, F. Boehme, C. L. Norman, M. Mace, and M. Gordon. What programmers really want: Results of a needs assessment for SDK documentation. In *Proceedings of the 20th Annual ACM SIGDOC International Conference on Computer Documentation*, pages 133–141, 2002.

[17] D. Pagano and W. Maalej. How do open source communities blog? *Empirical Software Engineering*, pages 1–35, 2012.

[18] D. Pagano and W. Maalej. User feedback in the appstore: An empirical study. In *RE*, pages 125–134, 2013.

[19] S. Palmiter and J. Elkerton. An evaluation of animated demonstrations of learning computer-based tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 257–263, New York, NY, USA, 1991. ACM.

[20] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[21] M. P. Robillard. What makes APIs hard to learn? Answers from developers. *IEEE Software*, 26(6):26–34, 2009.

[22] M. P. Robillard and R. DeLine. A field study of api learning obstacles. *Empirical Software Engineering*, 16(6):703–732, 2011.

[23] J. Singer, T. Lethbridge, N. Vinson, and N. Anquetil. An examination of software engineering work practices. In *CASCON'97*, page 21. IBM Press, 1997.

[24] T. Vestdam. *Generating Consistent Program Tutorials*, pages 55–70. 2002.

[25] R. Wallace, E. Soloway, J. Krajcik, N. Bos, J. Hoffman, H. E. Hunter, D. Kiskis, E. Klann, G. Peters, D. Richardson, and O. Ronen. Artemis: learner-centered design of an information seeking environment for k-12 education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, pages 195–202, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[26] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.