Standard Operating Procedures (SOPs) for Arabic Localization in Software Development Life Cycle - A Suggested Framework

رياض السلامين[1]* ليالي المزايدة[2] بلال القضاة[3]

تاريخ وصول البحث: 2023/05/14 م تاريخ قبول البحث: 2023/05/25م تاريخ نشر البحث: 2025/12/23م

Abstract

In the context of Software Development Life Cycle (SDLC), Standard Operating Procedures (SOPs) are guidelines that provide a framework for the development process. Applying the concept of SOPs on the SDLC may vary depending on the specific requirements of the project and the methodology used. It is important to have well-defined SOPs for each phase of the SDLC to ensure a smooth and successful software development process, as well as having been accepted by the end users and stakeholders of the developed solution. One key issue that needs to be highly considered during the SDLC for better user acceptance and successful solution delivery to the end user is the localization. Localization is highly impacting the business reach, brand recognition, and customer acceptance in the new markets with different cultural and linguistic nuances.

This research proposes a theoretical framework for Arabic cultural localization of the solutions during the SDLC, that is outlining a set of SOPs for a properly Arabic localized software solution, considering not only the language localization, but culturally as well. This modern approach is aiming to reach the Minimum Level of Acceptance (MLOA) among the Arabic users, by taking into consideration the main Arabic cultural issues as input for developing SOPs collected in a framework model, that can be generalized to be used by any developer who wants to easily reach the proposed threshold of acceptance in the Arabic region. This approach can be also considered as a set of standards for Arabic localization in software development.

Keywords: *Software Development, Arabic Cultural Localization, Software Engineering.*

إجراءات التشغيل القياسية (SOPs) لتعريب البرمجيات ضمن دورة حياة تطوير النظم: إطار عمل مقترح

الملخص

تُعد إجراءات التشغيل القياسية (SOPs) في سياق دورة حياة تطوير البرمجيات (SDLC) بمثابة مبادئ توجيهية توفر إطاراً منظماً لعملية التطوير. وقد يختلف تطبيق مفهوم هذه الإجراءات على دورة حياة التطوير بناءً على المتطلبات المحددة للمشروع والمنهجية المتبعة. ومن الأهمية بمكان وجود إجراءات تشغيل قياسية محددة بدقة لكل مرحلة من مراحل دورة حياة التطوير؛ لضمان سير عملية بناء البرمجيات بسلاسة ونجاح، فضلاً عن ضمان قبولها من قِبل المستخدمين النهائيين وأصحاب المصلحة. وتبرز مسألة "التعريب" (Localization) "كواحدة من القضايا الجوهرية التي يجب مراعاتها بعناية خلال دورة حياة التطوير لتحقيق قبول أفضل لدى المستخدم وضمان تسليم حلول برمجية ناجحة؛ حيث يؤثر التعريب بشكل كبير على الانتشار التجاري، وتعزيز العلامة التجارية، وقبول العملاء في الأسواق الجديدة ذات الفروقات الثقافية واللغوية المتنوعة.

[1] أستاذ مشارك، قسم هندسة البرمجيات، كلية تكنولوجيا المعلومات، جامعة الحسين بن طلال، معان، الأردن.

[2] أستاذ مشارك، قسم هندسة البرمجيات، كلية تكنولوجيا المعلومات، جامعة الحسين بن طلال، معان، الأردن.

[3] أستاذ مشارك، قسم هندسة الحاسوب، كلية الهندسة، جامعة الحسين بن طلال، معان، الأردن.

* الباحث المستجيب: reyad.m.salameen@ahu.edu.jo

يقترح هذا البحث إطار عمل نظري للتعريب الثقافي العربي للحلول البرمجية خلال دورة حياة تطويرها، وذلك من خلال تحديد مجموعة من إجراءات التشغيل القياسية الرامية إلى إنتاج برمجيات معربة عربياً بشكل سليم، مع مراعاة البعد الثقافي إلى جانب البعد اللغوي. يهدف هذا النهج الحديث إلى الوصول إلى "الحد الأدنى لمستوى القبول" (MLOA) لدى المستخدمين العرب، وذلك عبر أخذ القضايا الثقافية العربية الرئيسية كمدخلات لتطوير إجراءات تشغيل قياسية مجمعة في نموذج إطار عمل يمكن تعميمه واستخدامه من قبل أي مطور يرغب في الوصول بسهولة إلى عتبة القبول المقترحة في المنطقة العربية. كما يمكن اعتبار هذا النهج بمثابة مجموعة من المعايير المرجعية لتعريب البرمجيات في البيئة العربية.

الكلمات المفتاحية: *تطوير البرمجيات، التعريب الثقافي العربي، هندسة البرمجيات.*

1. Introduction

The SDLC is a methodology that is widely used in software development to guide the process of creating high-quality software that meets the requirements of the stakeholders. It refers to the process of developing software from conception to deployment (Umeugo et al. 2023). SOPs are a set of step-by-step instructions that describe how to perform a specific task or activity. In the context of SDLC, SOPs are guidelines that provide a framework for the development process, which can help organizations to improve the consistency, efficiency, and quality of software development.

The SOPs-based process helped to improve the quality and efficiency of software development. However, applying the concept of SOPs on the SDLC may vary depending on the specific requirements of the project and the methodology used. It is important to have well-defined SOPs for each phase of the SDLC to ensure a smooth and successful software development process, as well as having accepted by the end users and stakeholders of the developed solution (Gupta, 2021).

One key issue that needs to be highly considered during the SDLC for better user acceptance and successful solution delivery to the end user is the localization. It is clear that localization is a critical aspect among SDLC, that can help businesses to reach new markets, build brand recognition, and improve customer engagement. By adapting their content to suit the cultural and linguistic nuances of the target audience, businesses can avoid misunderstandings, comply with local laws and regulations, and create a more personalized and engaging experience for their customers. The localization of software solutions for the Arabic region or culture needs analyzing several related cultural and domestic issues. Successful localization of solutions for Arabic-speaking audiences requires a deep understanding of the local culture, social, customs, traditions, political landscape, as well as a willingness to make changes to the original content in order to ensure that it is respectful and appropriate for the target audience.

From there, the researchers found out the high demand of having a set of standards or SOPs that govern and facilitate the implication of Arabic cultural localization of the solutions during the SDLC. Accordingly; this research is outlining a set of SOPs for a properly Arabic localized software solution aiming to reach the Minimum Level of Acceptance (MLOA) among the Arabic users. This can be generalized to be used by any developer who wants to easily reach the proposed threshold of acceptance in the Arabic region.

2. Literature Review

SDLC is a systematic approach to developing software that involves a series of well-defined phases. The goal of SDLC is to ensure that software is developed efficiently, meets user requirements, and is delivered on time and within budget. It consists of several processes and phases that cover the lifetime of the software development journey. The most common phases of the SDLC are (Umeugo et al. 2023): requirements gathering, analysis, design, implementation, testing, deployment, maintenance: In this step, the software is maintained and updated to ensure that it continues to meet the requirements of the stakeholders.

SOPs are a set of documented procedures that outline the steps and activities required to complete a specific task or process. In the context of SDLC, SOPs can be used to ensure that software development is carried out in a standardized and consistent manner across different projects. The SOPs-based process helped to improve the quality and efficiency of software development (Gupta, 2021). However, applying the concept of SOPs on the SDLC may vary depending on the specific requirements of the project and the methodology used. It is important to have well-defined SOPs for each phase of the SDLC to ensure a smooth and successful software development process, as well as having accepted by the end users and stakeholders of the developed solution (Gupta, 2021).

Al-Qutaish (2009) discusses the development and implementation of SOPs for SDLC in an organizational setting. It provides an overview of the SDLC process and outlines the key activities and deliverables required at each stage. The paper also presents a case study of how SOPs were implemented in a software development project. The author also discusses the development and implementation of ISO standards for SDLC in small and medium-sized enterprises.

Lekh (2015) provides an overview of the SDLC process and outlines the key activities and deliverables required at each stage. The paper also presents a case study of how standards were implemented in a software development project and the benefits that were realized, such as improved quality and consistency. This shows that integrating of SOPs as kind of standards in SDLC in an organizational setting will be highly contributing to improving efficiency and productivity of the SDLC process and outlines the key activities and deliverables required at each stage and in enhancing project management and decision-making operations.

Localization is the process of adapting software or content to meet the linguistic, cultural, and functional requirements of a specific locale or market. It can be an important consideration during the several phases of the SDLC, especially when developing software for a global audience. In literature reviews, the topic of localization in SDLC has been extensively discussed, and there are many studies that examine different aspects of this topic. Some studies have focused on the localization process itself, exploring the challenges and best practices for adapting software to different cultures and languages. Others have examined the impact of localization on the overall SDLC, looking at how it affects development timelines, project costs, and user satisfaction (Gonzales & Zantjer, 2015).

Additionally, researchers have studied the role of localization in agile software development, exploring how agile methodologies can be adapted to accommodate the needs of localization teams and how localization can be integrated into the agile development process (Kabát, 2023). They also discuss the importance of communication and collaboration between developers and localization experts. While, Barakat (2017) explore the challenges of adapting open source software for different languages and cultures, and discuss strategies for involving the community in the localization process. They also discuss the importance of integrating localization into the development process to ensure that localized versions are maintained and updated alongside the original code.

From the reviews of the role of localization in software development, localization is important because it helps to ensure that the translated content is relevant, understandable, and engaging to the target audience. Gonzales & Zantjer (2015) mentioned that one of the key benefits of localization is that it allows businesses to reach a wider audience and expand their global reach. By adapting their content to suit the local market, businesses can increase their chances of success in new markets and improve their brand reputation. It also helps to improve the quality and accuracy of translated content. Translators who specialize in localization take into account the cultural and linguistic nuances of the target language, which helps to ensure that the translated content is accurate, appropriate, and culturally sensitive. Another important aspect of localization - as stated by Pym (2004)- is the use of appropriate language and tone. The tone and style of writing can vary significantly between different cultures, and it's important to use the right tone to communicate effectively with the target audience. This includes using appropriate levels of formality, avoiding slang or idioms that may not translate well, and using the right level of technical language (Lakó, 2012).

Also, Costales (2012) stated that software solution localization is not just about translating the text and voiceovers, but also about adapting the software to cultural differences. This can involve making changes to the software's graphics, software operation, and even storyline to ensure that it is appropriate and appealing to users in the target region. For example, a software that is set in a Western-style environment may need to be adapted to suit users in Asian markets. This may involve changing the characters' appearances, adjusting the software user experience, or even altering the storyline to better reflect the values and preferences of the target audience.

In addition, Costales (2014) mentioned that software solution localization also involves addressing technical issues related to language and culture. This includes ensuring that the software's text can display properly in languages with different character sets, adjusting the software's font sizes to accommodate different languages, and making sure that the software's voiceovers and sound effects are synchronized with the on-screen action. Another important consideration in software solution localization is the rating system used in different countries. For example, a software that is rated E (for Everyone) in the United States may be subject to different rating criteria in other regions, and may require additional changes to ensure that it is appropriate for the target audience.

Arabic region becomes an important target for most of the software manufacturers due the huge number of users and the very promising market share. So that, localizing software solutions for the Arabic region becomes very important for most of the solutions. However, localizing any solution for the Arabic region involves adapting the software to meet the specific cultural, linguistic, and technical requirements of users in the region. Accordingly; literature references provide valuable insights into the complexities of localizing software solutions for the Arabic region, and offer guidance on how to overcome the challenges involved (Abufardeh & Kenneth, 2008). These literature references provide further insights into the challenges and strategies involved in localizing software solutions for the Arabic region and offer valuable insights into the strategies and approaches that can be used to overcome these challenges.

Abufardeh & Kenneth's (2008) article discusses the challenges involved in supporting the Arabic language in software applications, particularly with respect to font rendering and text input. The author identifies issues related to different fonts, text direction, and the complexity of the Arabic language. The article concludes with recommendations for addressing these challenges in software localization efforts.

Omar et al. (2022) emphasizes the importance of cultural factors in software localization for the Arabic market. The author discusses issues related to user behavior, communication styles, and cultural norms that must be considered during the localization process. The article concludes with recommendations for integrating cultural considerations into software design and localization efforts. The article provides a systematic literature review of Arabic software localization, identifying common challenges and proposing strategies for addressing them. The authors identify challenges related to language, culture, and technical issues, and propose strategies such as user-centered design, cultural adaptation, and language testing to address these challenges.

Omar (2021) provides a review of the Arabic language and its support in software applications, including issues related to text input, font rendering, and keyboard layouts. The authors also discuss the challenges of supporting Arabic in programming languages and software development tools, and propose strategies for addressing these challenges. The author focuses on the usability of learning management systems in the Arab world. He identifies factors that can impact usability, such as language, culture, and technical issues, and propose strategies for improving usability, such as user-centered design and localization efforts.

3. Statement of the Problem

Localization solution requires a deep understanding of cultural differences, including language, humor, symbols, values, beliefs, religious and political sensitivities, legal and regulatory frameworks. By addressing these issues, developers can ensure that their solution is accessible, appropriate, and appealing to users in different countries and regions. Software localization solution needs to consider a wide range of cultural issues, including the portrayal of different cultures and ethnicities, gender norms, accents and dialects, historical and cultural context. In addition, software localization needs to consider User Interface (UI) and User

Experience (UX) design, including language, layout, color, graphics, software usage, and accessibility, which can help to establish a loyal user base and increase revenue.

This research aims to investigate the cultural issues that face the localizers of software solutions development specifically for Arabic culture, a relatively new topic that has received little attention by scholars. Successful localization of software for Arabic-speaking audiences requires a deep understanding of the local culture, social, customs, traditions, political landscape, as well as a willingness to make changes to the original content in order to ensure that it is respectful and appropriate for the target audience. However, localizing the software for Arabic-speaking users can present a number of cultural issues that must be addressed in order to ensure a successful and respectful developed product. By taking into account these factors and tailoring the solution to the specific needs and preferences of the target audience, developers can create engaging and culturally appropriate solution for Arabic-speaking users.

Most of the literature discussed the key factors that must be considered during localization, and presented their results only as related guidelines for considering localization during SDLC. However this research is proposing a set of SOPs to be followed during the SDLC, to ensure a successful Arabic localization for any software solution.

4. Methodology

This research will follow the following steps:

1. First, analyzing and identifying the major software cultural localization issues in general and Arabic culture localization in specific, that are mentioned in the literature references or collected from some of the domain experts.

2. Second, it is producing a proposed framework consisting of a set of SOPs for successful Arabic localization during SDLC, that can achieve the minimum level of acceptance (MLOA) among the software products' end users and stakeholders.

5. Discussion

The following are the main cultural localization issues that are identified by the researchers as main input variables for the suggested research's framework, which has been mentioned in O'hagan & Mangiron (2013), Onar & ÇATAK (2012) and other scholars:

- ♦ One common cultural issue in software solution localization is humor. What may be considered funny or appropriate in one culture may not be in another. Jokes and puns that work well in English may not translate well into other languages, and cultural references that are familiar to users in one region may not make sense to users in another.

- Another cultural issue in software solution localization is the use of symbols and imagery. For example, certain symbols may have different meanings in different cultures. In the Western world, a thumbs-up gesture is generally seen as a positive sign, but in some parts of the Middle East, it is considered offensive. Similarly, certain colors may have different associations in different cultures. In China, the color red is often associated with good luck and happiness, while in Western cultures it is often associated with danger or warning.

- Software solution localization also involves adapting the software to the cultural norms and values of the target audience. For example, software that features excessive violence or sexual content may need to be toned down or altered to comply with cultural norms in certain regions. Additionally, the software's storyline may need to be adjusted to reflect cultural differences in values and beliefs.

- One additional cultural issue that software solution localization may encounter is the use of language. Language can vary widely across different countries and regions, with differences in vocabulary, grammar, and syntax. This can make it challenging to translate a software accurately while still conveying the intended meaning. For example, in Japanese culture, there are specific honorifics and pronouns used when addressing different people, such as adding "-san" to someone's name to show respect. These nuances may not translate directly into other languages, and the use of different pronouns or honorifics in the translation could significantly alter the intended meaning of the original text.

- Religious and political sensitivity is another cultural issue that software solution localization needs to consider. Some countries or regions may have different religious beliefs or political views than the software's country of origin, and certain content may be considered offensive or inappropriate. Developers need to be sensitive to these differences and ensure that their software does not include any content that could be deemed insensitive or offensive.

- Software solution localization also needs to consider the legal and regulatory frameworks in different countries and regions. Some countries may have laws or regulations that restrict the use of certain content or require specific warnings or disclosures. Developers need to ensure that their software complies with all relevant laws and regulations to avoid any legal issues.

- Another important cultural issue in software solution localization is the portrayal of different cultures and ethnicities. Software solutions often feature characters from different backgrounds, and developers need to ensure that these portrayals are respectful and accurate. For example, software that features characters from different ethnicities may need to consider the cultural differences in appearance, behavior, and mannerisms. Stereotypical or caricatured portrayals of different cultures can be offensive and insensitive and can harm the software's reputation and sales.

- Similarly, software solution localization needs to consider the gender norms and expectations of the target audience. Some countries may have different gender roles and expectations than others, and developers need to ensure that their software is respectful of these differences.

- Another cultural issue is the use of regional accents and dialects. In some cases, the use of a specific accent or dialect can add authenticity and depth to a software's characters and setting. However, accents and dialects can also be difficult to translate accurately and may not make sense to users in other regions.

- Software solution localization needs to consider the historical and cultural context of the software's setting. Software that are set in historical periods or feature historical events need to be accurate and respectful of the cultures and traditions of the time. Inaccurate or insensitive portrayals can be offensive and insensitive to users in those regions.

- Another important consideration in software solution localization is the user interface (UI) and user experience (UX) design. UI and UX design need to be tailored to the target audience's preferences and expectations, including language, layout, and navigation. For example, the layout of the UI may need to be adjusted to accommodate different character sets and text directionality. Some languages, such as Arabic and Hebrew, are written from right to left, which can require significant changes to the software's UI design. Similarly, the use of color and graphics in the UI may need to be adjusted to appeal to the target audience's preferences. Different cultures may have different color associations, and some graphics may not be familiar or appealing to users in certain regions. Localization of UX design may also involve adapting software user experience to suit the target audience's preferences. For example, some regions may prefer fast-paced, action-oriented software UX, while others may prefer slower-paced, strategic software UX. Developers need to be aware of these differences and adjust the software's mechanics accordingly to ensure that users in different regions find the software enjoyable and engaging.

- Finally, software solution localization may also need to consider accessibility issues for users with disabilities. Developers need to ensure that their software is accessible to users with visual, auditory, or motor impairments. This may involve adding alternative text descriptions, closed captions, or audio cues to the software.

The researchers then identified the following main Arabic cultural localization issues as stated by Al-Batineh & Alawneh (2022), Al-Batineh (2021), Al-Ajarmeh & Al-Adwan (2022) and other scholars; Arabic cultural issues may pose significant challenges when localizing software solutions. Localizing software solutions for Arabic-speaking audiences can present a number of cultural challenges that developers need to be aware of. Some of the key cultural issues that must be considered include:

- Religious sensitivities: Islam is the dominant religion in many Arabic-speaking countries, and certain aspects of the religion, such as depictions of the Prophet Muhammad, are considered taboo. Software developers must ensure that their software do not include any content that is disrespectful or offensive to Islam or any other religion.

- Language: Arabic is a complex language with many dialects, and it is essential to use the appropriate dialect when localizing a software for a particular country or region. Developers must also ensure that the Arabic translation of the software is accurate and culturally appropriate.

- Gender roles: In some Arabic-speaking countries, there are strict gender roles that dictate how men and women should behave. Developers must be careful not to include content that is perceived as disrespectful or offensive to women, such as sexualized or overly aggressive female characters.

- Violence and gore: In some Arabic-speaking countries, depictions of violence and gore are considered taboo. Developers must be careful not to include excessively violent or graphic content in their software.

- Cultural references: Many software solutions contain references to popular culture, such as music, movies, and television shows. Developers must be careful to include references that are appropriate and relevant to the target audience, and avoid references that may be unfamiliar or offensive to Arabic-speaking users.

- Social norms: Social norms in Arabic-speaking countries can differ significantly from those in other regions of the world. For example, software that depict alcohol consumption or drug use may not be well received by users in these countries.

- Censorship: Some Arabic-speaking countries have strict censorship laws that prohibit certain types of content, such as political themes or explicit material. Developers must ensure that their software comply with local censorship regulations in order to avoid legal issues and make the software available to users in those countries.

- Regional differences: It is important to note that Arabic-speaking countries are not homogenous, and there are significant differences in culture, language, and customs between different countries and regions. Developers must take into account these regional differences when localizing their software, and tailor the software to the specific needs and preferences of the target audience.

- Historical and political sensitivities: The Middle East has a complex history of political and social turmoil, and certain topics, such as political conflicts or religious extremism, may be sensitive or controversial. Developers must be careful not to include content that may be perceived as politically or culturally insensitive or offensive.

- Local market factors: In addition to cultural factors, developers must also take into account local market factors, such as pricing, distribution, and competition from other software in the region. It is important to tailor the software to the specific needs and preferences of the target audience, while also considering the practical and economic realities of the local market.

- Localization of text direction: Arabic is a right-to-left language, which means that the direction of text and other elements in the software must be adjusted to suit the local language and cultural norms. Developers must ensure that all text and other elements in the software are localized correctly, and that they are aligned and formatted correctly for right-to-left reading.

- Voice acting and audio localization: Voice acting and audio localization are important elements of software localization, and must be adapted to suit the local language and cultural preferences of Arabic-speaking users. This can include the use of local accents and dialects, as well as appropriate music and sound effects that reflect the cultural identity of the target audience.

As examined before in the literature, most of them discussed the key factors that must be considered during localization, such as language, culture, and technical requirements. They mainly presented their results only as related guidelines for considering localization during software development project phases such as: planning, design, implementation, testing, and maintenance.

6. Proposed Framework

Based on the major issues that are mentioned in the literature references that covers this topic and the ones that were collected from some of the domain experts, the research is proposing SOPs that guarantee a successful Arabic localized software development. They are selected in a consistence manner and formulated into a theoretical framework so that the research's framework could then be generalized for whom who would like to develop any Arabic-localized software as step-by-step checklist.

The key SOPs that are presented by this research would be:

1. **Conduct research**: Before localizing a software solution for Arabic-speaking audiences, it is important to conduct thorough research into the target culture. This may involve studying local customs, traditions, and taboos, as well as gaining an understanding of the local language and social hierarchies.

2. **Consult with local experts**: It is essential to work with local experts, such as language professionals, cultural consultants, and software testers, who can provide insight into the nuances of the Arabic language and culture. These experts can also help to identify potential cultural issues and suggest appropriate changes to the software content.

3. **Evaluate the software content**: Software developers and localizers should carefully evaluate the software content to identify any potential cultural issues. This may involve reviewing the software script, character designs, and in-software text to ensure that they are respectful and appropriate for the target audience.

4. **Make necessary changes**: Based on the research and consultation with local experts, software developers and localizers should make necessary changes to the software content to address any cultural issues. This may involve altering character designs, changing dialogue or in-software text, or removing certain elements altogether.

5. **Test the software**: After making changes to the software content, it is important to thoroughly test the software to ensure that it is culturally appropriate and engaging for the target audience. This may involve working with local testers to identify any remaining issues and make necessary adjustments.

6. **Provide cultural context**: In some cases, it may be helpful to provide additional cultural context within the software, such as through in-software tutorials or support resources. This can help to educate users on local customs and traditions, and enhance their overall experience of the software.

7. **Consider regional differences**: Arabic-speaking countries have different dialects and cultural nuances, and it is important to take these differences into account when localizing software solutions. Software developers and localizers should consider which regions they are targeting and tailor the software content accordingly.

8. **Use appropriate language**: The Arabic language is complex and nuanced, and it is important to use the appropriate language when localizing software solutions. Software developers and localizers should use proper grammar, avoid slang and colloquialisms, and use language that is appropriate for the target audience.

9. **Be aware of religious sensitivities**: Religion is an important part of many Arabic-speaking cultures, and it is important to be aware of religious sensitivities when localizing software solutions. Software developers and localizers should be respectful of religious beliefs and avoid content that could be seen as disrespectful or offensive.

10. **Engage with the local community**: To better understand Arabic culture and the target audience, software developers and localizers should engage with the local community.

This may involve attending local events, working with local influencers, or conducting surveys to gain insights into local preferences and cultural nuances.

11. **Incorporate local elements**: To enhance the user experience, software developers and localizers should consider incorporating local elements into the software. This may involve incorporating local landmarks, food, or cultural traditions, or incorporating local music or art into the software design.

12. **Respect local laws and regulations**: Different countries have different laws and regulations governing the content of software solutions, and it is important to respect these laws when localizing software solutions for Arabic-speaking audiences. Software developers and localizers should be aware of local laws and regulations and ensure that the software content complies with these regulations.

13. **Use appropriate character designs**: When designing software characters, it is important to use appropriate designs that are respectful of Arabic cultural norms. This may involve avoiding revealing clothing or suggestive poses, as well as ensuring that character designs are diverse and representative of the target audience.

14. **Consider local holidays and events**: To enhance the user experience and create a sense of cultural immersion, software developers and localizers should consider incorporating local holidays and events into the software content. This can help to create a more engaging and relatable experience for users.

15. **Address cultural stereotypes**: Software solutions often rely on cultural stereotypes to create characters and settings, but these stereotypes can be offensive and harmful. Software developers and localizers should be aware of cultural stereotypes and work to avoid them in the software content, or to subvert them in a positive and respectful way.

16. **Provide appropriate support and documentation**: To ensure that users can fully engage with the software content, software developers and localizers should provide appropriate support and documentation, such as in-software tutorials, FAQs, and support resources. These resources should be localized and tailored to the needs of the target audience.

17. **Work with local voice actors**: To ensure that the software is properly localized and resonates with the target audience, software developers and localizers should work with

local voice actors to provide voiceover work for the software. This can help to create a more authentic and immersive experience for users.

18. **Test for cultural sensitivity**: After making changes to the software content and incorporating local elements, it is important to test the software for cultural sensitivity. This may involve conducting focus groups or surveys with local users to identify any remaining cultural issues and make necessary adjustments.

From the above examined steps, fig.1 below shows the framework that is proposed by this research that can be generalized as standardized SOPs for software developers and manufacturers to ensure a successful Arabic culture-considering solution, which would have a significant market reach and wider user acceptance among in the very promising Arabic markets.

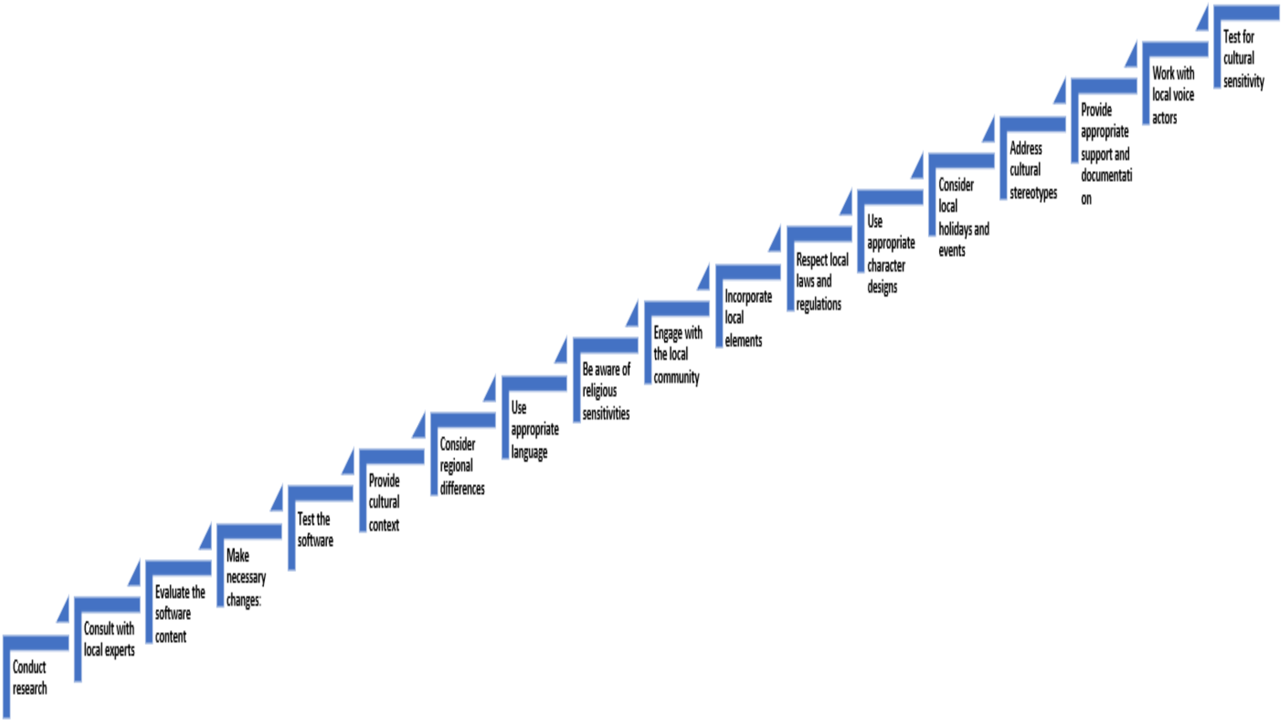
Conduct research
Consult with local experts
Evaluate the software content
Make necessary changes:
Test the software
Provide cultural context
Consider regional differences
Use appropriate language
Be aware of religious sensitivities
Engage with the local community
Incorporate local elements
Respect local laws and regulations
Use appropriate character designs
Consider local holidays and events
Address cultural stereotypes
Provide appropriate support and documentati on
Work with local voice actors
Test for cultural sensitivity

Figure 1: Research Proposed Framework: SOPs for Arabic Localized Software

7. Conclusions

Accordingly, the research is proposing a theoretical framework that is outlining a set of SOPs for a successful Arabic-localized software solution, that is properly set to cover all the issues that are needed to be considered during the life cycle of the development process, considering not only the language localization, but culturally as well, in order to generalize outlined standards for Arabic localization in the SDLC that can achieve the MLOA among the software products' users and stakeholders.

By following these proposed SOPs, developers will be effectively considering a wide range of Arabic culture localization issues during the software development, and create software that are engaging, immersive, and culturally respectful for the targeted audiences. The methodology adopted in this research to generate the mentioned research's framework, is the analysis and experimentation on the main Arabic localization issues in the software solutions, and accordingly identifying the key actions -as SOPs- that would mitigate these issues and yield to specific threshold of acceptance for the developed software.

8. Future Work

A potential future work proceeding to this research could be to study some of the main attempts of Arabic localization in some famous software solutions and try to investigate the compliance of these selected sample of solutions with the outlined standards that are proposed by the research's framework. The solutions can be analyzed and compared against the proposed framework of this research.

References

- Abufardeh, S. & Magel, K. (2008). Software localization: The challenging aspects of Arabic to the localization process (Arabization). In proceedings of the IASTED International Conference on Software Engineering, SE 2008, (pp.275-279).

- Al-Ajarmeh, O., & Al-Adwan, A. (2022). Insights into blending game localisation in the Arab world: Arafiesta as a case study. The Journal of Internationalization and Localization, 9(1), 61-84.

- Al-Batineh, M. (2021). Issues in Arabic video game localization: A descriptive study. Translation & Interpreting, 13(2), 45-64.

- Al-Batineh, M., & Alawneh, R. (2022). Current trends in localizing video games into Arabic: localization levels and gamers' preferences. Perspectives, 30(2), 323-342.

- Al-Qutaish, R. E. (2009). Measuring the Software Product Quality during the Software Development Life-Cycle: An International Organization for Standardization Standards Perspective. Journal of Computer Science, 5(5), 392-397.

- Barakat, S. (2017). Arabic localization of open source software-A case of open journal system. Asian Journal of Information Technology, 16(7), 626-631.

- Costales, A. F. (2012). Exploring translation strategies in video game localization. *MonTI: monografías de traducción e interpretación*, (4), 385-408.

- Costales, A. F. (2014). Video game localisation: adapting superheroes to different cultures. *Quaderns: revista de traducció*, 21, 0225-239.

- Gonzales, L. & Zantjer, R. (2015). Translation as a user-localization practice. Technical Communication, 62(4), 271-284.

- Gupta, A., Rawal, A., Barge, Y., (2021). Comparative Study of Different SDLC Models. International Journal for Research in Applied Science and Engineering Technology. 9 (11), 73-80.

- Kabát, M. (2023). Possible Translation Problems, Their Causes, and Solutions in Agile Localization of Software. International Journal of Interactive Mobile Technologies (iJIM). 17(1). 129-140.

- Lakó, C. (2012). Localization and Translation Studies. *Studia Universitatis Petru Maior. Philologia*, 12, 202-208.

- Lekh, R. (2015). Exhaustive study of SDLC phases and their best practices to create CDP model for process improvement. In 2015 International Conference on Advances in Computer Engineering and Applications. pp (997-1003).

- O'hagan, M. (2005). Multidimensional translation: A game plan for audiovisual translation in the age of GILT. Challenges of Multidimensional Translation, 76-87.

- O'hagan, M., Mangiron, C. (2013). Game localization. Translating for the global digital entertainment industry. John Benjamins Publishing Company.

- Omar, A., Altohami, W. Ethelb, H. & Hamidi, B. (2022). Localization Quality Assessment for More Reliable E-Commerce Applications in Arabic. Education Research International. 2022. 1-14.

- Omar, A. (2021). An Evaluation of the Localization Quality of the Arabic Versions of Learning Management Systems. International Journal of Advanced Computer Science and Applications. 12(2). 443-449.

- ONAR, B. C., ÇATAK, G. (2022). Translation and localization of video games: An analysis of uncharted 4 in Turkish. Connectist: Istanbul University Journal of Communication Sciences, 62, 91-122.

- Pym, A. (2004). Localization from the perspective of translation studies: Overlaps in the digital divide. In *Katmandu: Scalla Conference.* pp*(*1-7).

- Umeugo, W., Lowrey, K., Pandya, S. (2023). Effect of SDLC Models on The Perception of SSDLC Innovation Characteristics and SSDLC Adoption Intention. International Journal of Security (IJS). 14 (1), 1-16.