Jalmari Kunnari

Localization Pipeline for a AAA Game Production

Bachelor's Degree in Business Information Technology

Bachelor of Business Administration

Spring 2021

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Jalmari Kunnari

Työn nimi: Lokalisaatioprosessi AAA pelituotantoon

Tutkintonimike: Tradenomi, tietojenkäsittely

Asiasanat: peliala, pelinkehitys, pelituotanto, lokalisaatio

Tämä opinnäytetyö on toteutettu toimeksiantona nimeltä mainitsemattomalle pelialan yritykselle, joka kehittää AAA-pelejä, eli suuren budjetin ja kehitystiimin vaativia videopelejä. Toimeksiannon tehtävä on määrittää lokalisaatioprosessi kyseisen yrityksen tarpeisiin osaksi suurempaa tuotantoprosessia. Lokalisaatioprosessi tämän opinnäytetyön kontekstissa tarkoittaa prosessia, jota noudattamalla videopeli saadaan lokalisoitua sen tuotannon aikana. Tämä lokalisaatioprosessi on määritelty nimenomaan kyseisen asiakasyrityksen kehittämisympäristöön, mutta se on myös tarpeeksi universaali, jotta se on tarpeen vaatiessa sovellettavissa samankaltaisiin käyttötapauksiin.

Opinnäytetyön ensimmäisessä pääkappaleessa käydään läpi muutamia pelituotannon ja lokalisaation peruskäsitteitä. Peliala on edelleen suhteellisen nuori verrattuna muihin viihdealoihin, joten on olennaista selittää eräitä peruskäsitteitä liittyen pelien tuotantopuoleen tämän opinnäytetyön kontekstissa. Toinen puoli tästä kappaleesta keskittyy lokalisaation käsittelyyn, ja etenkin mitä se voi tarkoittaa pelituotannon yhteydessä.

Toisessa pääkappaleessa perehdytään itse yritykselle laadittuun lokalisaatioprosessiin. Prosessi on jaettu kolmeen osaan, esituotantoon, tuotantoon sekä jälkituotantoon. Nämä ovat sitten entisestään jaoteltu pienempiin välivaiheisiin. Kaikkia kolmea tuotantovaihetta havainnollistetaan kulkukaavioilla, jotka kuvaavat lokalisaatioprosessin kulkua sen alusta loppuun. Kaavion lisäksi kaikki tuotantovaiheet ja niiden välivaiheet käydään läpi tekstimuodossa. Kaaviot ja teksti tukevat toisiaan muodostaen kokonaisuuden, joka on itse lokalisaatioprosessi.

Kolmannessa pääkappaleessa käydään läpi merkittävimpiä riskitekijöitä koko lokalisaatioprosessin kannalta, samalla perehtyen siihen millä toimenpiteillä niiltä voidaan mahdollisesti välttyä. Tämän lisäksi tarkastellaan kevyesti kuinka toteutettavissa nämä toimenpiteet ovat tämän opinnäytetyön tilaajayrityksen kehitysympäristössä.

Neljännessä ja viimeisessä pääkappaleessa käsitellään itse lokalisaation merkittävyyttä osana pelituotantoa. Tämä kappale sisältää myös tekijän omaa pohdintaa sekä havaintoja lokalisaatioprosessista. Yksi olennainen asia on se, miten lokalisaatioon voidaan suhtautua tilaajayrityksen kaltaisissa kehitysympäristöissä.

Abstract

Author(s): Jalmari Kunnari

Title of the Publication: Localization Pipeline for a AAA Game Production

Degree Title: Bachelor of Business Administration, Business Information Technology

Keywords: games industry, game development, game production, localization

This thesis has been carried out as a commission to an unnamed game development company that develops AAA video games. The goal of this thesis was to define a localization pipeline for the company's development purposes as part of bigger production process. Localization pipeline for those that are not familiar with the term means a step-by-step process, when followed, a video game can be localized during its production. This localization pipeline has been designed specifically for the customer company, but it is made in a way that allows it to be adapted for similar kind of use cases if needed.

In the first main chapter some of the key concepts in game production and localization are explained. Game industry is still a relatively young when compared to other fields of entertainment, so it is appropriate to go through some of its key concepts related to production of games in the context of this thesis. The other half of this chapter will in turn focus on doing the same for localization.

The second main chapter will go in detail explaining the devised localization pipeline. The pipeline is split into three parts, which are pre-production, production and post-production. These three production phases are also further broken down into smaller sub-stages. All of the three production phases are visualized as flowcharts, which depict the pipeline progression from start to finish. Alongside the flowcharts all of the production phases and the sub-stages are explained more in detail in a text format. The flowcharts and text portions together in tandem form the pipeline itself.

In the third main chapter some of the main risk factors in the pipeline are laid out. At the same time, it is explored what kind of countermeasures can be employed to help mitigate these risk factors. In some cases, it is also briefly thought how feasible these measures are within the customer company.

In the fourth and final main chapter the authors thoughts and conclusions on the whole subject will be presented. Especially, what kind of significance localization holds as a part of video game production. This includes questions like how localization itself is viewed in environments similar to the customer company.

Table of contents

List of Symbols

AAA Term used for video games, that have large, usually multi-million-dollar budget and development team consisting somewhere around hundred or more developers.

Locale Locale, or locale version, means a localized version of a game that has text and VO elements translated into some other language than the games source language.

LQA Short for localization quality assurance, which is the same as QA but focuses solely on localization related bugs and issues.

QA Short for quality assurance, which means testing and playing the game in different stages of the development to map out and document various bugs and issues.

VO Short for voice-over, which means audio elements in the game that play when characters in the game are talking.

1 Introduction

In the AAA game industry, pipelines are something of a necessity. For game productions the size of several hundred developers and with development times minimum of two years, it is essential to have processes in place on how to proceed with various production tasks.

The main goal of this bachelor’s thesis is to define a localization pipeline for a customer company that produces AAA video games. Since this pipeline is made as a commission, it means that the work towards this thesis is a case specific process, which is made for this particular development studio. However, that does not mean that at least some parts of the pipeline could not be utilised by other similarly operating studios. Currently at the customer company, there is no process written down for localization and how it should be conducted. This thesis has been conducted by documenting the current relevant practices that the development team at the customer company is already doing and getting acquainted with literature on game localization and -production. In addition, during the writing of this thesis, the author has discussed about the related topics with his colleagues at the customer company.

There are already written books such as “Border Crossings” by Gambier and Doorslaer or “Translation and Localization Project Management: The art of the possible” by Keiran Dunne and Elena Dunne, which go through some of the best production and project management practices when it comes to localization. The aim of this thesis is to review some of those practices and recognise certain common issues that should be avoided during the production of a game. Furthermore, to be clear on the point of view, the author will be talking about the process of localization management from game producer’s standpoint. The range of addressed issues will also be strictly limited to those that are related to the localization process itself.

It is likely that some of those suggested practices are not necessarily viable in an environment for what this work has been conducted for. Because of that, the aim is also to try to effectively filter out practices that would not be suitable in this working environment. The practices should be viable in a way that they can benefit AAA game projects, which are meant for traditional retail markets. With that the author is referring to the types of games that are produced in a more traditional way in today’s game industry. The production has a clear start and an end, which results in a published game product that has completed its development cycle. This type may include DLC packages that are produced as an extension for the base game, expanding on the original content. This will leave out any mobile platform and live service developments, which are looking to be more and more common in today's game industry.

Also, when talking about localization in the games industry there are two slightly different processes that fall under that term. The first and most common meaning for the term localization is translation of text and VO elements present in the game. And the second one is modifying the game content itself to be compatible with certain target markets. For example, some things that might be okay to present in the game in western world, where it is developed, might be considered offensive or insulting somewhere in Asia. The most notable example in this case is China, where something like appearance of skeletons or skulls needs to be removed altogether.

In this thesis however, only translation related processes are considered, since content modification in a forementioned way has been a fraction of the total work put into localization in the company and it is very likely to remain like that in the near future. In addition, generally it is the translation work that cover most of the localization related work.

According to the author's personal observations, pipeline as a term is still not that widely known outside of game- and software development. At least not exactly in a same way that the concept is understood and used in video game production. It was also noticed that sources talking about these kinds of pipelines were sparse. That is why the next few chapters will go through the key areas of this thesis, including pipelines, based on already established theory.

In the second main chapter of this thesis, the devised localization pipeline will be laid out, which is the main work of this thesis. It is defined by using the author's personal experience accumulated by working at the customer company, gathering feedback from more experienced colleagues, and researching the topics of this thesis.

Then, the third main chapter will go through some of the more commonly known pitfalls and issues that might undermine any production efforts to produce quality localizations. The topics mentioned in that chapter were identified mostly by working on localizations within the customer company and focusing on those was thought to be the right approach by the author for this thesis.

Lastly, in the fifth and final main chapter the author will lay out some of his thoughts and conclusions about the whole subject of the thesis. How localization can be viewed by the development team and the importance of it, are things that are thought about.

1.1 Localizing different types of game content

There are generally a few different categories of content, which will get localized in games. Text and audio being the most common ones, which is why this thesis is going to be mainly addressing those two. Other types include things like in-game textures that contain text and any other type of game content that might need to be modified to be suitable for certain target markets.

1.1.1 Text localization

Compared to localizing any other type of content, the process for texts is the most straightforward and easily understood. If the game has any kind of readable text in it, it will need to be localized. Most of the texts in the game can be grouped under a term “OST” (on-screen-text). “…’on-screen text’, in-game text refers to all the text present in the user interface (UI) (such as menus, help messages, tutorials and system messages), narrative and descriptive passages, and all dialogues that are not voiced-over and only appear in written form…” (O'Hagan & Mangiron, 2013.) Also, along with the OST there are dialogue related subtitles that are included in the text localizations.

Texts that are planned to be localized are usually stored in their own string files. Strings are individual text entries in the game. They can be as short as a single word present in menus or multiple lines of dialogue that contain several sentences, including time stamps for subtitle timings. To some degree it depends on the narrative- and gameplay teams how these are organized. The strings themselves are in the company’s case stored in multiple XML-files. This allows the developers easily view and edit the contents of these files using various software tools.

It is mainly because of the forementioned reasons why text localization is the easiest one to conduct. Even if there are no content management tools used for handling the text files, they are simple enough to work with as they are. The files can be passed to a localization vendor without any additional preparation and if formatting is being paid attention it is a very low risk effort. But if content management tools are used, it only makes everything much better as they offer features like detailed version history. More on the content management software is written in chapter 2.3.

1.1.2 Voice over localization

In voice over localization, the original in game voices, and most importantly, the dialogue is re-recorded in the target locale language. If the game contains songs, sometimes those can be also localized if there is a strong need for it. (O'Hagan & Mangiron, 2013.) Voice over localization is significantly bigger undertaking than text localization as it involves casting a new set of actors for each target locale, recording of the materials, and editing them. Depending on the arrangements for VO localization, all of the forementioned steps can be taken care by the localization vendor chosen for the project. In some cases, this process can be made slightly easier by having a single actor to voice multiple character. This is usually considered mainly for non-main cast characters.

Voice over localization always requires that the text localization has already been at least partially completed and casting for the source language characters has been conducted.

1.1.3 Localizing other game content

There are some additional types of content that might be localized alongside text and VO. Some of the texts within the game can be incorporated in a various in-game textures that are latched on different 3D-objects. These can be in-game maps, signs or some other environmental pieces that would naturally have text in them. (O'Hagan & Mangiron, 2013.) These textures are images as themselves, which means they would need to go through significantly different workflow than for example what the text assets go through. This also means that they cannot be run through any content management software mentioned in this thesis, which the author will explore briefly in chapter 2.3. Also, considering resources, the art assets naturally would require someone like graphic designers' involvement in order to make sure that everything fits together when translating for example from English to German. Especially with those languages, change in the text dimensions would be pretty drastic. This is most likely one of the key reasons that art asset localization is not practiced as much as text or VO localization, as it would require more significant efforts to make it work, while there are not too many benefits that would outweigh the required efforts. In some cases, it might even be a thematic choice from the development team not to localize these types of assets, for example, due reasons based on narrative design of the game.

O'Hagan and Mangiron (2013) suggest however, that these types of materials should always be localized. The reality is that in some cases the developers might simply choose not to since the localization process for them is much more complicated. O'Hagan and Mangiron (2013) are presenting an idea that the assets should initially be designed in a way that it would make them easier to be localized in the later stages of the production. One such option is to build up the textures in a way that the text used in them are stored separately in a text file like all the other strings.

However, if in-game signage happens to be a critical to player guidance, there are requirements from certain game console manufacturers to have these translated in one way or another. One solution for this is to simply display translated text on the screen when player is looking at certain signs.

1.2 Pipelines

Investopedia.com describes pipelines in the following way, which matches very accurately how it is used in the game development: "The pipeline metaphor is often used to describe progress through a series of stages culminating in a long-term goal. In many cases, the phrase is used to describe an ongoing process." (Fernando, 2020.)

But in the context of this thesis, it is also important to define pipeline more in context of game development. "The game development pipeline is the process of building a video game from concept to completion. Much like a production line, the pipeline helps organize the flow of work so that everyone knows what they need to deliver and when." (Stefyn, 2019.) Along with helping to organize project structure, using pipelines will also help managing the game budget, which in turn can help highlights inefficiencies and bottlenecks. This is said on an assumption that the required pipelines are defined and developed early enough for a given game production to be able to benefit from them.

As stated earlier, in a nutshell the main purpose of pipelines in game development is to clarify work processes that are running during game production. They will help to manage the timeline and the budget of a given project, reducing inefficiencies and bottlenecks. While pipelines vary between projects and studios, according to Stefyn (2019), the process should be more or less the same in AAA- or in mobile development, and also be structured in a way that allows flexibility and possibility to change course if needed. A game is something that is constantly evolving throughout the development process, new features and content being developed and added all the time. Due to this so some processes that seemingly work on paper might not do so in reality. "Therefore, the pipeline is not necessarily a linear process. Work must be sent for creative approvals and can often be sent back for revisions. Pipelines must be flexible enough to factor revisions and course changes.” (Stefyn, 2019.)

Pipelines are usually visualised as flowcharts of varying designs, which aim to make them clear to interpret even by individuals who might not be that familiar with the related project. Colours, shapes, and additional formatting are used to clarify and highlight connections between different stages of the pipeline. However, there are as many different ways of visualizing pipelines as there are people who are designing them. While emphasising the visual presentation would be the best approach in terms of comprehension, for some it may lack a necessary level of detail, which something like pure text format offers. And of course, there can be something in between, like the model presented in this thesis. The next picture (Figure 1) offers an example of a typical pipeline flowchart.

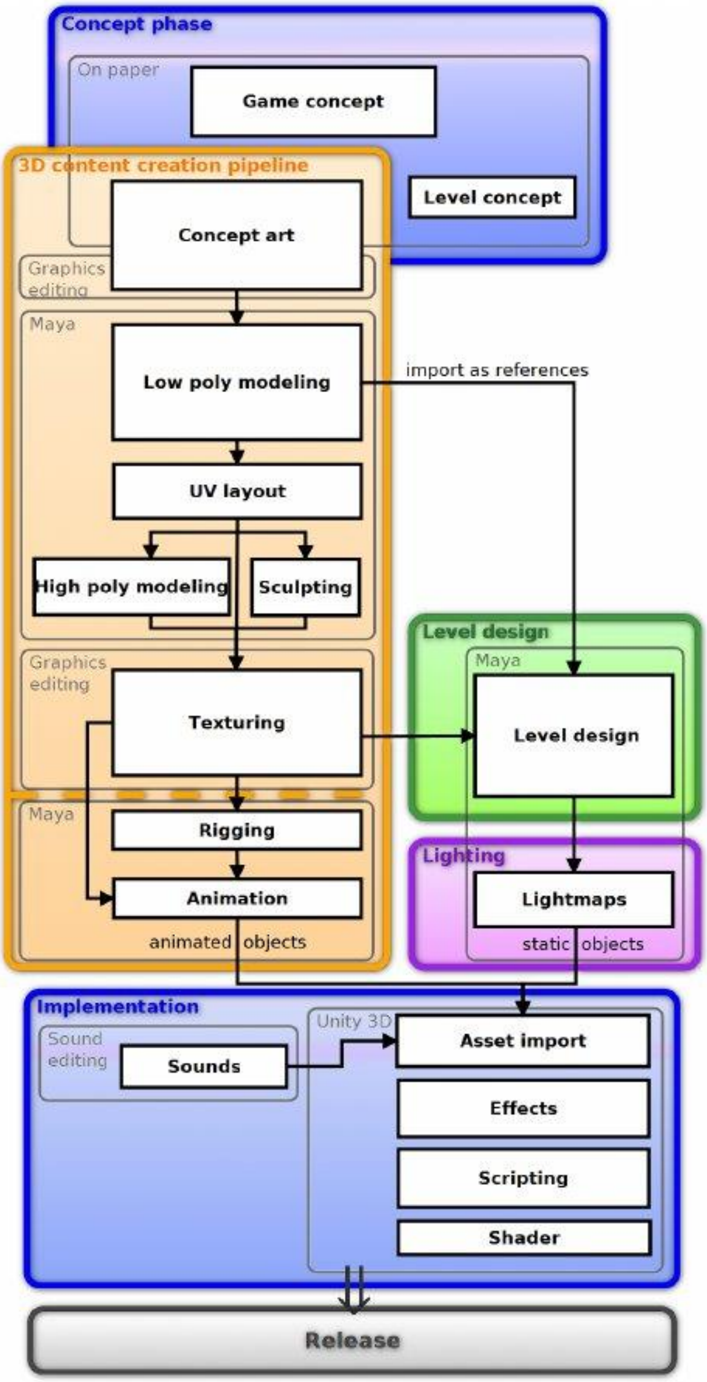

Figure 1. Example of high-level game production pipeline (Labshutz et al., 2011.)

1.3 Content management software for localization

In some companies, game strings are still managed by using only Excel files, manually sending files containing massive tables to localization vendor through an e-mail or FTP (file transfer protocol). In bigger game companies, especially ones which work in the field of AAA, where the amount of game strings can be counted in tens of thousands, it is greatly beneficial to use some of the content management tools specifically designed for game localization.

The key benefit of using a content management software is that it allows an access to all of the strings in a same place. The software displays the source language strings as well as the locale versions side to side, who edited them and when, and if they are still unlocalized or the translations are out of date. They also have a feature to modify the strings in the software itself, so that the development team can address minor issues regarding the strings on their own without having to ask the localization vendor for help.

The customer company is currently using content management software called Localize Direct. Localize Direct is relatively simple content management software made for localization purposes, which can offer great benefits when used in bigger projects. The software is used by the development team only to manage the locale strings. It supposedly also has features to handle transferring and storing audio assets, but that option has not been explored in the past and the current methods used to handle those are sufficient enough. How the software is used is that the company’s IT department maintains a server database where the string files are uploaded by using in-house tools. Localize Direct is then used by the localization producer to view the contents of the database.

Additionally, in the company one other content management software has been investigated and evaluated, which potentially could be a better option for Localize Direct. Owned by Keywords Studios, which is a company specialized in providing a multitude of different game development related services, XLOC offers their own namesake software. It differs from Localize Direct in a few ways. Where Localize Direct is its own software, XLOC is much more accessible by being a web-application. XLOC also offers much more options in terms of displaying the content and language options, which already make it better feature wise.

2 Localization pipeline

This chapter of the thesis goes through explaining the three stages of the devised localization pipeline. The three stages are pre-production, production, and post-production. These are the general stages of any production operations. The production stages are then further broken down to a set of sub-stages, which the pipeline as a whole is comprised of. The pipeline itself is made up of flowcharts and text parts. The flowcharts are used to visualise the pipeline and to make it more easily comprehensible and then the text parts go more in detail explaining what all of the different sub-stages mean in practice. Pipeline flowcharts 1, 2 and 3 represent parts of the pipeline in pre-production, production, and post-production, respectively. It is designed to be read from top to bottom. The sub-stages are presented as boxes in the flowcharts and flow through them is displayed with arrows. Colour coding indicates what part of the development team is supposed to take care of which sub-stage. Figure 2 on page ten, clarifies what color represent which part of the development team. Responsibility over some of the sub-stages can be split between various parts of the development team so some of the boxes can be filled with more than one color. In the flowcharts, percentages are used to display how big of a time emphasis each of the production stages should approximately have. All of the three production stages equal to one hundred percent and that number is divided between the production stages.

Also, there are few sub-stages that are marked in the flowcharts but, which are citations from other production pipeline that maps out sub-stages for the narrative team. These outside sub-stages are acting here in the localization pipeline as dependency indicators for some of the localization sub-stages.

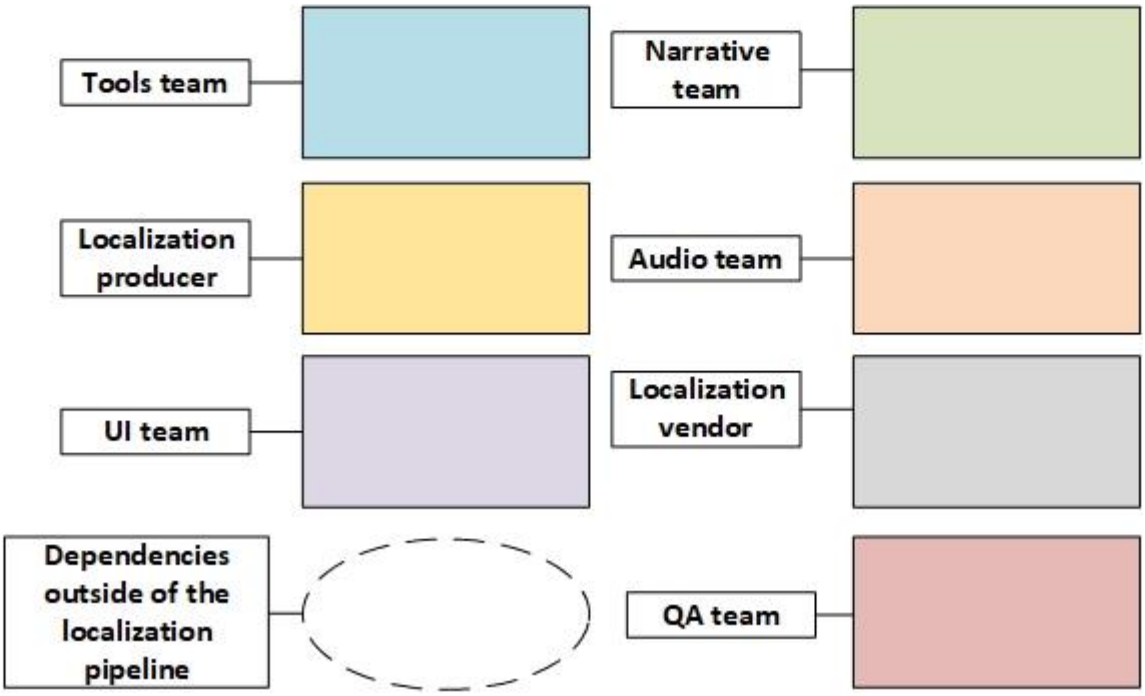

Figure 2. Pretext for colors and shapes used in the pipeline flowcharts

The actual production structure for a given project might not follow the production stages exactly as described here, but the pipeline should be adaptable with some additional planning. Also, it is fair to mention that even though the given project would be following the pipeline structure with exactly the same project structure in place, it would be realistic to assume that some of the steps might bleed into the next upcoming production stage due to various reasons.

Regarding the sub-stages presented within the pipeline, these have been validated by having them reviewed by the author's producer colleagues and by checking on what other industry professionals are saying about localization processes. For example, Mark Timmins, who has around ten years of experience in video game localization, points to very similar steps in his blog post on Gamasutra.com. (Timmins, 2018.)

The localization pipeline of this thesis is in many aspects similar to a localization model called "Simship localization", which is mentioned in "Translation and Localization Project Management: The art of the possible" by Keiran Dunne and Elena Dunne. "Simship localization" refers to a model where a release of the locale versions of the game are done at the same time with the source language version. That is achieved by developing the source language version of the game and the locale versions simultaneously. One major aspect is that the planning for the localizations is done at the same time the source language version is being developed. This includes collaboration between the development team and the localization vendor before the actual translation of the materials start.

2.1 Pre-production

Time in pre-production is exclusively dedicated to planning the localization process. All of the different factors that might affect the localization process should be mapped during this time. The main factors include the localization budget, the number of locales for text and VO, LQA arrangements, localization vendor, publisher- developer relations etc. If the game has a separate publisher, they might have some preferences regarding some of these factors.

Regarding the time that will be spent on a project in total, pre-production for localization should require around 20 percent of the total production time. Figure 3 displays the pipeline flowchart for the pre-production phase.

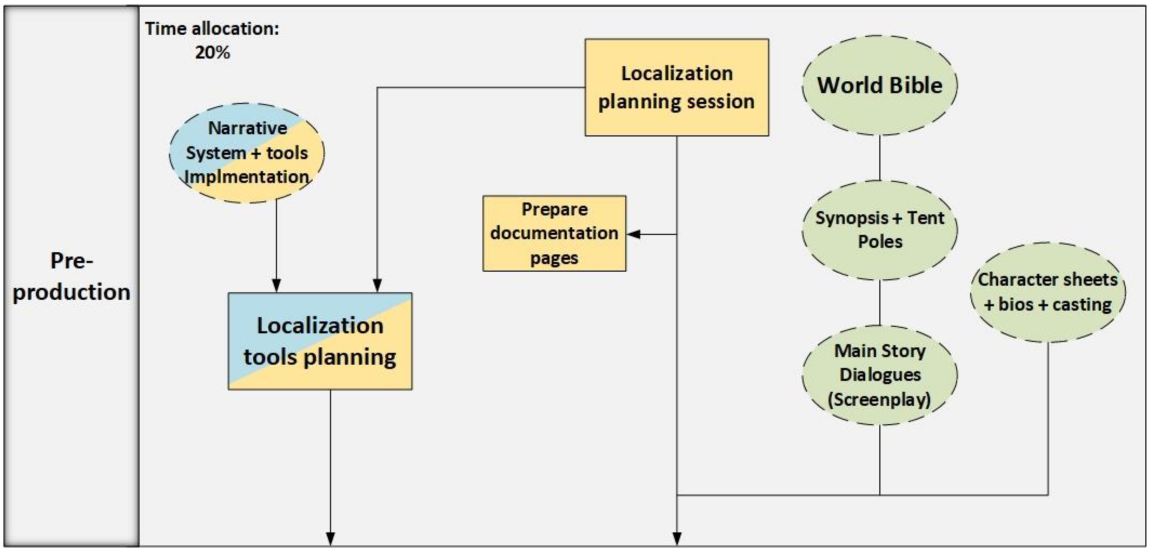
Pre-production
Time allocation: 20%
Narrative System + tools Implmentation
Localization planning session
World Bible
Prepare documentation pages
Synopsis + Tent Poles
Character sheets + bios + casting
Localization tools planning
Main Story Dialogues (Screenplay)

Figure 3. Pipeline flowchart 1, pre-production

Localization planning session: Localization producer needs to have a discussion with the publisher of the game and agree on some localization related matters. This can be a series of discussions and meetings, but the main goal for these is to get answers to at least the following questions.

- What is the overall scope of the localization efforts?
- What is the number of locales and which languages will be included?
- What asset types will be localized?
- Which localization vendor will be used?
- Does the publisher have a preferred vendor?
- How is the LQA going to be conducted?
- How will the localization related costs be covered?

In the customer company's case publishers have usually had a say in localization related matters across their various projects.

Localization tools planning: Regarding the tools planning, it is sensible to re-evaluate the tools used in the previous localization operations. Pre-production is also the most suitable occasion to do any evaluation for old or new tools, to see if something is worth replacing to improve production quality. The more later this is done the more it can compromise not only the production quality, but also any related timelines and budgeting.

In the customer company's case, all of the general systems have already been put in place for text and VO asset workflows, so the main question is if the current content management software is something that the development team wants to continue using. Additionally, regarding the in-house tools, it could be worth exploring if there are any new features that the tools team could implement to make the localization process better. For this sub-stage, **Narrative System + tools implementation** is a dependency existing outside of the localization pipeline that needs to be addressed before planning for localization tools can properly be started.

Prepare documentation pages: Using the main documentation platform in the company, it would be advisable to create documentation pages for the upcoming localization process. On these pages should be documented all of the things agreed with the publisher, as well as any pipelines or workflows related to the localization process.

There are few additional outside dependencies in the pre-production of the localization pipeline (Figure 3), which are on the narrative team. **World Bible**, **Synopsis + Tent Poles**, **Main Story Dialogues (Screenplay)**, and **Character sheets + bios + casting** need to be all completed by the narrative team before moving on to the production phase of the localization pipeline.

2.2 Production

Production is the latest time to plan anything localization related. Preferably most of the planning has already been conducted in pre-production, and then in production, preparations are being made according to those plans. In production it is time to set up the framework on which the asset localization workflow is going to be happening. During production some of the materials can already be sent to the localization vendor to be translated if certain criteria are achieved by this time.

Regarding the time that will be spent on a project in total, pre-production for localization should require around 40 percent of the total production time. Figure 4 displays the pipeline flowchart for the production phase.

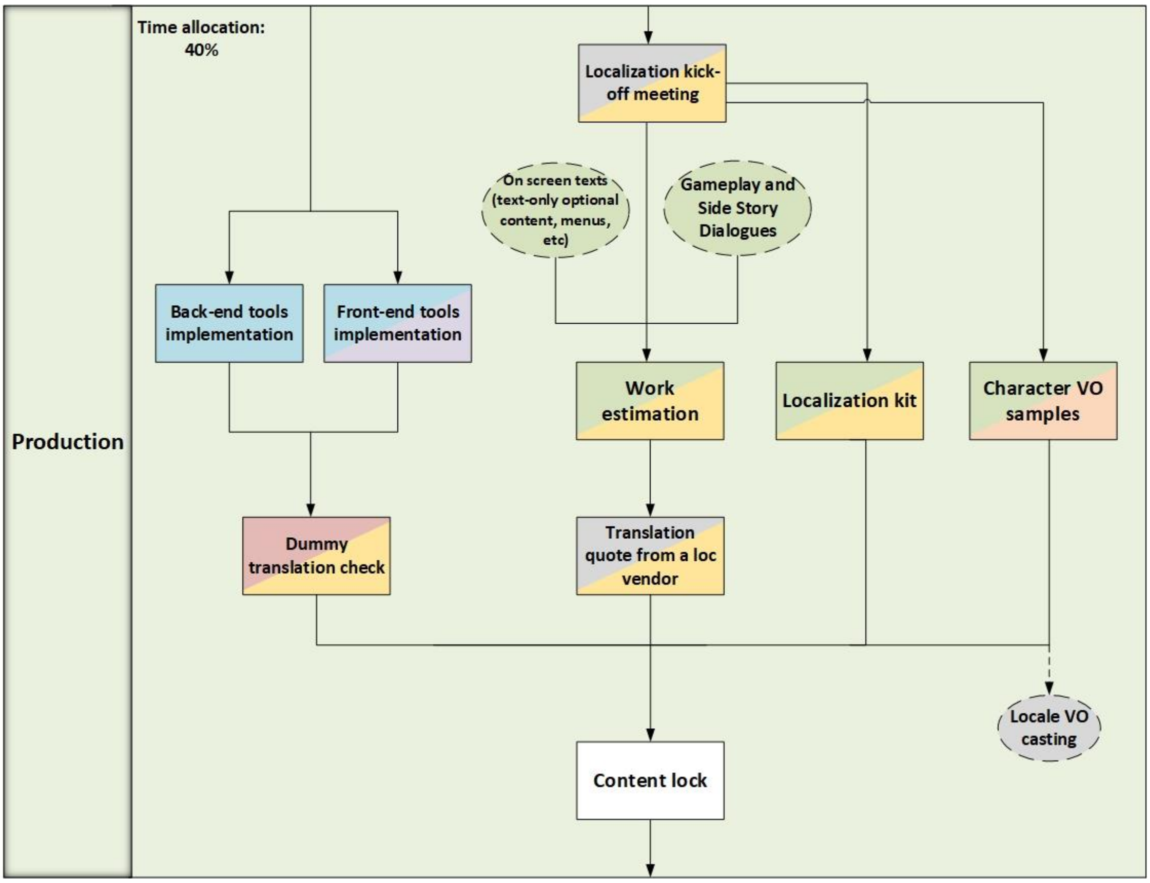
Production
Time allocation: 40%
Localization kick-off meeting
On screen texts (text-only optional content, menus, etc)
Gameplay and Side Story Dialogues
Back-end tools implementation
Front-end tools implementation
Work estimation
Localization kit
Character VO samples
Dummy translation check
Translation quote from a loc vendor
Locale VO casting
Content lock

Figure 4. Pipeline flowchart 2, production

Front-end tools implementation: This step requires the localization producer to create a new project in the used content management software and have someone to do the required configurations on the backend side, so that all of the required software work in sync. This will be explored more in chapter 4. By the latest, at this point the localization producer needs to be granted an access to the project repository and create a required workspace for himself so he can run the asset implementation workflow.

In order to have all of the required languages displayed properly in the game there needs to be fonts for them. By default, the engine should handle and be equipped with systems and fonts for western standard Latin alphabet. In a very likely case where the game is localized, for example, to any Asian language, the game engine requires fonts that can display those required Asian languages. Generally, the games art director in cooperation with the UI team will think about and agree on the fonts for the source language, after which UI team is usually responsible of implementing them. But regarding fonts that might be required specifically for some of the locales, acquiring them might fall onto the localization producer. In some cases, the games publisher might be providing certain required fonts.

Back-end tools implementation: Back-end tools implementation in this case mean the configuration between the used version control, the content management software and any in-house tools required. In the company's case it is a task for a tools programmer who is the most familiar working with these software setups.

Dummy translation check: After the necessary tools have been implemented and made sure that all of the required fonts have been acquired, a dummy translation check needs to be performed. In this step both the tools implementation and fonts will be tested. Localization producer needs to run the tools workflow for the content management software with pseudo translations generated by the content management software itself, if it supports that kind of feature, or Google Translate for example. This should be done for all of the text language locales that are planned for the game.

Localization kick-off meeting: Once the localization vendor has been chosen, the relevant people from the development team need to have a meeting with the key personnel from the localization vendor. Topic for this meeting would be to discuss about the overall nature of the project. During this it would be good to try to figure out approximate timeline when the actual localization work would be happening, including when things like localization kit and the word counts should be delivered by the development team.

Work estimation: So that the localization vendor can make a realistic cost estimate of the upcoming localization work, detailed word counts are needed for both text and VO elements of the game. Word counts need to be calculated by the narrative team as accurately as possible and due to that, this sub-stage is dependent on two additional steps that exist outside of the localization pipeline. **Gameplay and Side Story Dialogues** and **On screen texts (text-only optional content, menus, etc.)**, which both need to be completed by the narrative team.

VO localization also requires some additional details how the content is categorised. Additionally, narrative team needs to specify which of the spoken dialogue lines either do not have time constraints, do have time constraints, or do have strict time constraints. For ones that do not have time constraints it does not matter if the localized audio is any amount of shorter or longer than the original audio. Then for the ones that are marked having time constraints, the localized audio file needs to comply with pre-set time frame according to the length of the original audio file. For these the development team can define that the localized audio file can fall short on or exceed the original length with certain amount of seconds. But then for the ones that are marked as strictly time constrained, the localized audio file needs to be very close or exactly the same length as the original one.

Translation quote from a localization vendor: Before any localization work can start, there needs to be a work quote form the localization vendor so that contract for the work can be defined. This is always something for the localization producer to handle in collaboration with the project's executive producer.

Localization kit: Assembling a localization kit is probably the single most important step that will affect the quality of the translations. Localization kit is a comprehensive info package from the development team to the translators, which should include all the relevant information the translators need in order to come up with the most suitable and fitting translations for the game. All of the needed components of the localization kit are well defined in Damien Yoccozs blogpost on Gamasutra.com. (Yoccoz, 2019.) Yoccoz is a founder of Level Up Translation, a company that offers app, board- and video game localization services. With this he has good understanding what translators want to see in the localization kit. Those things are condensed into following six points.

1. The first one is to specify for what kind of audiences the game is aimed at. Nowadays different types of players are categorized into varying groups depending on their age, gender, how seriously they engage in video games etc. Specifying what kind of demographic the game is aimed at, will help the translators to decide exactly what kind of language should be used in the translations.

2. The second one is to define what kind of formatting the translated materials need to follow. It is mentioned that information about punctuation style, capitalization preferences, preferred abbreviations and any other similar kind of preferences should be included.

3. The third point is to describe the tone of the game. "Is your game supposed to be funny and sarcastic? Or is it rather dark and gritty? Maybe it's more conversational, or rather formal in tone. Localization will vary drastically according to the tone you specify. There are endless ways to localize a given string, depending on the tone you choose." (Yoccoz, 2019.)

4. The fourth one is to provide comprehensive character introductions. With this it is crucial to remember that the translators want to and need to know everything the developers themselves know about the characters.

5. The fifth point is to describe and provide details about the world the game takes place in. Similar to the previous point it is important to let the translators in on all of the details the development team knows about the game world.

6. The sixth one is to provide as much context as possible. In some of the languages it is important to know details like if certain people mentioned in the text are male or female

Something that is not mentioned in Yoccozs list but should be included somewhere in the deliverables meant for the localization vendor, are technical limitations that certain strings might need to follow. For example, if the game has some kind of collectible items that are supposed to display a certain amount of text on a certain amount of screen space, details for this kind if boundaries need to be provided. The strings that need to follow these limitations should have a pre-defined number of how many individual characters those strings can contain. This is done in order to prevent the text in the translated strings from bleeding over the designated screen space dedicated for it.

Format for the localization kit could be something like a pdf-document, but if for example the used content management software supports documentation that could be used similarly, that kind of solution could be also used.

Character VO samples: Alongside delivering the localization kit, VO samples for all of the in-game cast should be delivered to the localization vendor. This is needed in order for the localization vendor to conduct casting for all of the VO locale versions. The localization vendor should be provided this as early contractually possible, since casting and arranging the recording sessions are one of the more time-consuming phases of the localization process.

Locale VO casting: This step is mentioned here as a possibility depending on the arrangements for the project. Even though locale VO casting is conducted by the localization vendor, sometimes it could be beneficial for the development team to be involved in the locale casting. This topic is explored more in chapter 4.

Content lock: Content lock should conclude the production part of the localization pipeline. At this point all of the narrative and gameplay content should be locked so that there would not be any changes to it beyond this point. However, if changes after this are required for whatever reasons, they should be aimed to be kept in minimum.

At this point of the pipeline all of the source language strings should be exported to the localization database with a help of the established software tools setup. For the strings, the author has assembled a workflow for the customer companys case (Figure 5).

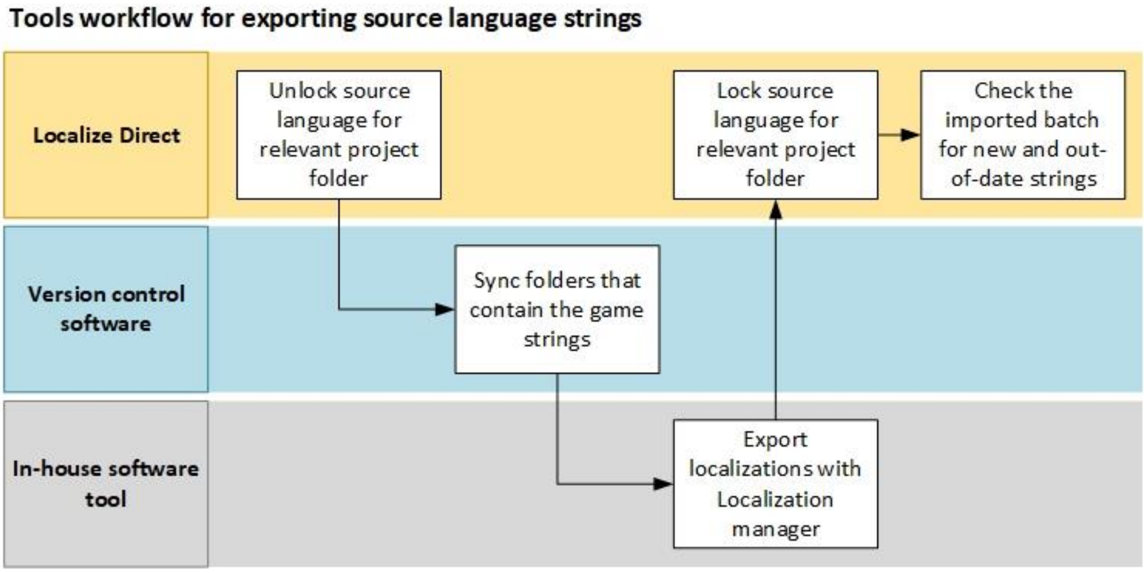
Tools workflow for exporting source language strings
Localize Direct
Version control software
In-house software tool
Unlock source language for relevant project folder
Sync folders that contain the game strings
Export localizations with Localization manager
Lock source language for relevant project folder
Check the imported batch for new and out-of-date strings

Figure 5. String export tools workflow

The forementioned workflow applies only for the strings so the VO files are handled a bit differently. In the customer company's case, the source language VO files are stored in pre-established FTP repository, where they can be accessed and downloaded by the localization vendor.

2.3 Post-production

In a sense post-production is the quietest and also at the same time the busiest time for localization. Thas is because part of the time in post-production is spent waiting on the translations to be completed and other part resolving the issues after the localizations have been implemented. Naturally, post-production is also the period where all of the remaining materials left to be localized are sent to the localization vendor.

Regarding the time that will be spent on a project in total, pre-production for localization should require around 40 percent of the total production time. Figure 6 displays the pipeline flowchart for the post-production phase.

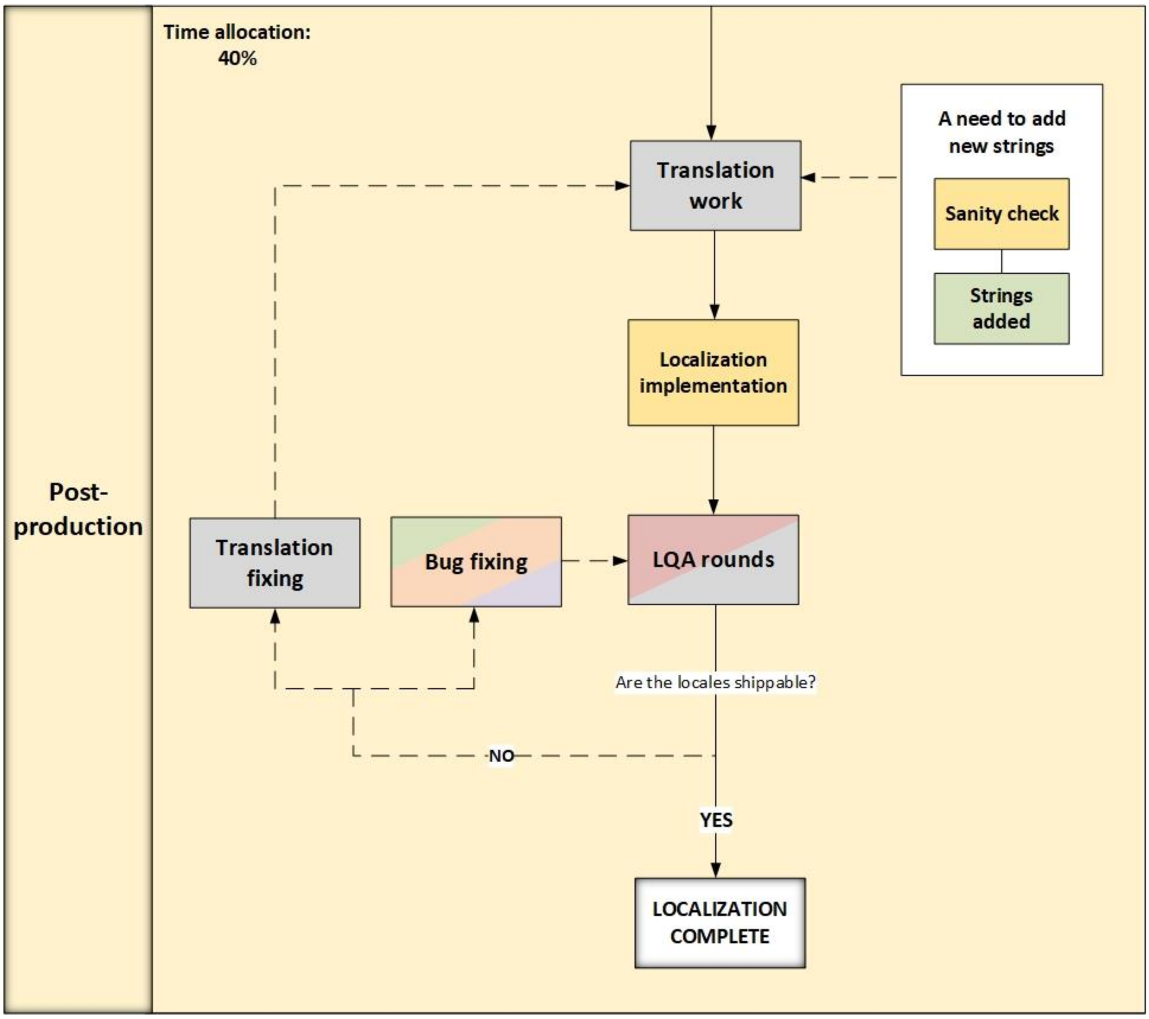

Figure 6. Pipeline flowchart 3, post-production

Translation work: In the translation work, all of the assets that require localization have been delivered to the localization vendor and they are working on the translations, locale VO recordings etc. Sometimes depending on the arrangements, the assets can be delivered and worked upon in different sized batches. Reason to this might be for example, if some of the materials are still waiting to be finalized by the development team. In this case it would be a good option to deliver all of the materials that are ready and have the localization vendor start the localization process partially. How long this step will take depends solely on the amount of localized materials and a capability of the localization vendor.

If at any point in the post-production there is a need to add new string to the game, they need to go a short approval cycle. First, the localization producer needs to do a **Sanity check** on the number of new strings if it is in any way feasible at this stage of the production. If the new strings can be added without compromising the projects other timelines the new strings can be added to the existing translation work.

Localization implementation: When the localization vendor is finished localizing all of the assets or a batch of them, the next step is to implement them into the game. Different companies have their own workflows explaining how this is done, since version control and software used can vary between companies. For this thesis' purposes the author has assembled a workflow for the customer companys case (Figure 7).

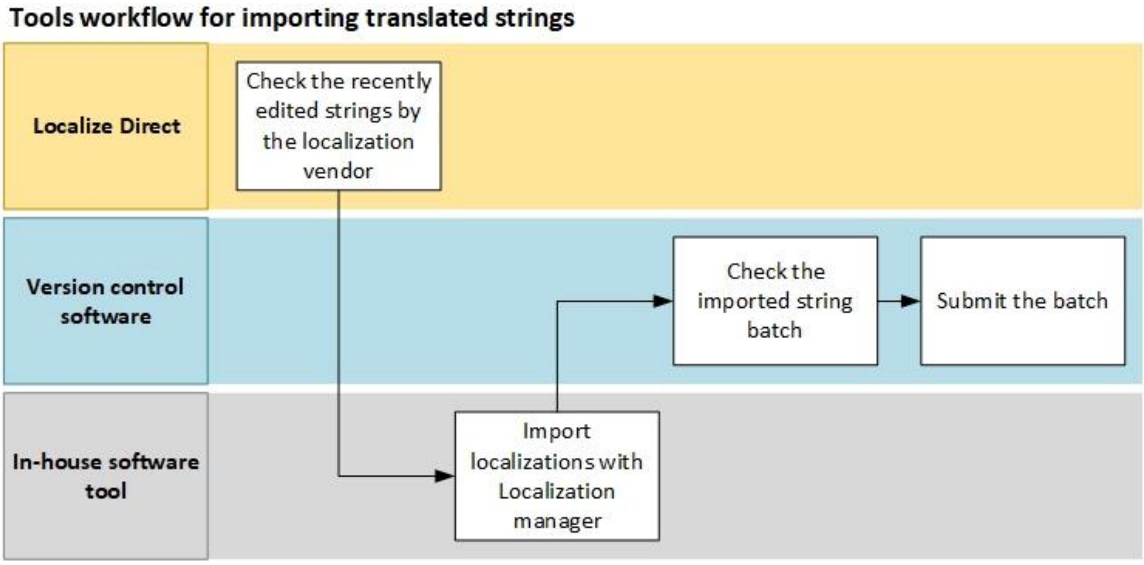
Tools workflow for importing translated strings
Localize Direct
Check the recently edited strings by the localization vendor
Version control software
Check the imported string batch
Submit the batch
In-house software tool
Import localizations with Localization manager

Figure 7. String import tools workflow

The forementioned workflow, like in the case of exporting, applies only for the strings and the VO files are again handled differently. The locale VO files are stored by the localization vendor in the same FTP repository as before, where they can be accessed and downloaded by the development team. After acquiring the VO files, the localization producer needs to submit them to a correct repository and file folders in the company's version control software.

LQA rounds: After the localized materials have been implemented into the game and a game build that includes them has been produced, it is time to have that build tested. Localization quality assurance can be handled separately from the typical QA personnel by outsourcing it to external vendors who are specialized in that kind of testing. The duration of this step is also heavily dependent on few different factors, like for example which of the locales are getting tested and if the testers are playing through the whole game or only certain parts of it.

After the LQA rounds, what will be done next depends on if the locales are deemed shippable by the development team. If the answer is no, the next step will be to try to fix the issues and bugs reported during LQA. If the answer is yes and the development team deems that the locales are good enough to be shipped completing the localization process will be the next step.

Bug/translation fixing: After LQA rounds have been completed or whenever bugs related to the localizations have been discovered, they are reported in the used project management software like Jira. If there are any changes to be made to the locale strings, those will by default go back to the localization vendor to be fixed and the pipeline loops back to the **Translation work** sub-stage. Then, if the issues are about anything else than the locale strings, they will be assigned to the development team and the pipeline eventually loops back to the **LQA rounds** sub-stage.

Localization complete: When this sub-stage is reached the localization for the game should be completed as the locales are deemed shippable. However, after this point, the developer can agree on maintaining some level of development support for the locales. For example, it is a possibility that some major issues with the locales are discovered only after the game has been released.

3 Points of concern in the pipeline

In this main chapter the author is addressing some elements that should be given some thought in the localization process. These are things that, if neglected, might lead to issues of varying magnitude in the pipeline.

3.1 Dependencies

Localization is something that is affected by different dependencies as it is located at the very end of the overall game production. Dependencies mean that various steps in the localization pipeline cannot be started unless at least one other step in another pipeline has been finished. In the pipeline flowcharts 1 and 2 some of these dependencies are pointed out.

In order to localize anything, there needs to be material that translators and other production personnel can work upon. The localization process is mostly dependent on a couple of disciplines, which are the narrative, gameplay and audio teams and depending on the project the amount varies between these teams. Even though the narrative team produces most of the text materials in the game, depending on the studio, the gameplay team might be responsible of coming up with some of the strings that are closely tied to gameplay or even UI. But even if majority of the gameplay or UI related texts are coming from the gameplay team, they should be proofread by the narrative team before submitting them to further down the pipeline.

It is also the narrative team that is mostly responsible of assembling the localization kit and coming up with the word counts for the localization vendor. Even if all of the content would be ready to be localized, it would not be possible to start that work without these two deliverables. The narrative team needs to be allocated the time to come up with both the localization kit and the word counts some time before the localizable content is locked.

VO elements are the other major part of the materials. VO recordings and editing for the source language needs to happen before the localization process for them can start. Depending on the planned schedule, in theory, casting for locale VO could be started as soon as the source language casting has been concluded.

In both of the cases of the text and VO materials, the main question is if there is a point doing the localization work in batches or wait until all of the source materials have been finished and one hundred percent locked. If there is enough commitment from the development team to stick with the edit locks for the content, it could be a viable choice.

3.2 Budget

Budget is something that is going to determine a scope for several things in the localization process, affecting heavily how the rest of the points in this chapter can be addressed. Generally, the best option would be to localize all the text and voice-over elements in the game and have them run through proofreading and LQA. But sometimes due to budgeting reasons, the scope of these actions might be constricted.

The following things are the main factors, which dedicate how much budget the localization process is going to require. The main thing is how much materials there is to be localized. After that comes how many locales there will be for text and VO. The VO localization can be done only for the main cast in some cases to minimize the budget. Then, how is the LQA going to be conducted and how extensively is another big factor for the budget. Finally, the number of content delivery batches to the localization vendor might affect the budget a bit. It also depends on how the statement of work is defined for the localization vendor.

Depending on the project, localization budget might be a thing that is determined by the game's publisher, or maybe the development studio itself if the game is going to be self-published. Publisher usually also makes an educated decision to how many different languages the game is going to be localized, since they usually have much better understanding of the different target markets. The amount of languages directly affects how much the localization is going to cost. The selection of languages might depend on things like if the game is a sequel in a series where the previous games have been localized in a certain set of languages. In this case it could be favourable to at least go for the languages used in the previous titles. Then there is the proofreading and LQA. These are important parts of the pipeline and depending how these are arranged will affect the budget significantly. That is why these topics should be brought up when there are going to be talks about localization in the project.

3.3 Tools integration

Like stated previously in the earlier chapters of this thesis, tools integration should be handled in the pre-production-, or early in the production stage of the project. This is to ensure that any issues that might come up are dealt with well before any assets are planned to go through the related workflows. Tools integration in a context of the localization pipeline presented in this thesis means making the chosen content management software, related in-house software tools and the version control system to operate in tandem. This is being done to facilitate the handling of the localized assets, most notably the strings.

Without going too much into technicalities in the context of this thesis, the most likely place where issues regarding tool integration will most likely occur, is the link between the in-house tools and the content management software. In the customer company's case, the in-house tools are the component doing most of the lifting in this equation. It pushes the original source language files to the localization database, which is set up by the customer company's IT department. With this setup any updates made to the in-house tools have a chance to create various issues, which for example might prevent exporting or importing of the strings from the localization database. Then if there are any issues with the servers the database is located in, that can prevent the development team, the localization vendor or both from accessing the strings altogether. Some of these issues are highly likely to pop up at some point during the development and there might not be any realistic pro-active measures that can be applied to prevent them from happening.

3.4 Translation quality

There are a number of things, which can potentially affect the quality of the translations. The very first thing affecting the translation quality is a quality of the source language materials. This is a bit of an obvious one, but of course the translations can only be as good as the source materials provided.

The localization kit should be the most significant pre-emptive measure to ensure better translation quality, but after that there are a couple of factors that cannot be made that much better even with a good briefing.

One additional very important thing is how in tune the translators are with the context of the game. For example, if the game in question is a military shooter type of a game, it would make sense for the translators to have some kind of basic understanding of military lingo and its usage. This is a topic that could be brought up in the localization kick-off meeting with the localization vendor, for example if the game has a strong setting. Otherwise, it is usually the localization vendors responsibility to select the most suitable translators for a given project.

Then, when getting into the translation work itself, probably the most difficult thing for the translators to work on is humour. Not only is humour subjective, but when translating jokes, puns and word plays into another language, many things can be lost in translation. Some things might be so much tied to the source language that in order to produce meaningful localizations, some phrases might need to be re-invented altogether. "…this may be observed in the adaptation of Batman Arkham Asylum into Spanish, where there are a number of puns and riddles that cannot be effectively transferred into the target language without suffering a partial or a total loss in meaning." (Costales, 2012.) If the game contains a lot of jokes and puns or otherwise have jokes in places where the narrative team wants to indicate something specific with a given line, the team should think about some alternatives or just generally be prepared that some of the lines are not able to be adapted properly.

Some sources like "Border Crossings" by Gambier and Doorslaer state that to proactively combat against poor translations, the translation team should be integrated to the development team itself during production in some way or another. This would definitely be a beneficial thing for the translation quality if, for example, there would be one translator per language to analyse and do some groundwork on the initial source materials.

One issue with that this kind of approach is the budget. The earlier the translators are being involved with the project the more expensive the localization process will naturally become. This kind of method might also feel like unnecessary work, since game development tends to be very volatile in a way that changes to the source materials might be very abrupt and come late in the production. It is not too uncommon that in the late stages of the development the content goes through big overhauls. For this reason, the author deems it would be the best to start the translation process after the source language materials have been locked and most preferably polished.

3.5 Locale VO quality

VO quality for the locales can be broken down to a few elements. How well the actor suits their character and how well they understand and portray the character. Then there is of course the quality of recordings and editing. With these, there are these few possible things which can bring down the quality of the locale VO even on its own.

Alexander Murauski on his blogpost has devised a list containing nine points (Murauski, 2016), if followed might alleviate these possible quality issues with the VO localization. Murauski is the Chief Operational Officer of Alconost, a company specialised in offering localization services.

1. His first point is to pay attention to project formatting, which will mostly benefit the recordings and editing. How subtitle timings are sequenced, marked and who is doing that work.

2. Second point is to provide detailed explanations of the characters that are to be voiced, describing their personalities, and providing pictures that display their character. This is information that should be included in the localization kit by default.

3. Third point on his list is to limit the number of voices, which is already mentioned in chapter 3. Reducing the number of different voice actors does not only make the localization process cheaper, but also simpler. One risk with this approach is that the in-game voices might end up sounding too similar between different characters, but this risk can be reduced by a skilful actor and editor.

4. Fourth point recommends having only one voice per audio file. Even though some in-game scenes might involve multiple characters talking to each other, having one file for one speaker makes editing and arranging the audio in the game engine less prone to risks.

5. Point number five mentions about making a pronunciation guide to supplement the game script. "Some rules of pronunciation are generally accepted, but others must be determined by the company itself. Your studio should study the script and make a list of all the words whose pronunciation may present discrepancies." (Murauski, 2016.)

6. Sixth point on the list might be a more difficult one to conform to, which is to leave space in scenes. This is to help accommodate any additional length that the different language

translations might present. Especially German is one of those languages, that when translated, it usually always ends up increasing in length.

7. Point seven on Murauskis list to provide plenty of source materials. The bare minimum is of course to provide the original audio files of all the recordings, but Murauski is recommending to also provide videos of the instances where the audio is used. If the game includes pre-rendered cinematic cutscenes, videos of those should be provided to the localization vendor if possible.

8. Point number eight is to ensure that the provided script of the game matches exactly the other materials provided. This might prove to be an issue if the game content has not managed to get locked early enough in the localization pipeline. Even if relatively small discrepancies are noticed, it may fuel bigger doubt if the materials that are being worked upon are all up to date.

9. Ninth and the last point recommends ensuring that the people involved in translating the materials, which are used for the audio recordings, are native speaking professionals. With more established localization vendors this should be self-evident, and this point should be only of concern if the development studio is looking into something like crowdsourcing to help with their localization efforts.

During the casting for the source language VO, to ensure the best possible quality, it would be beneficial to have the development team in some form to review the casting options for the locales. The localization vendor for the project handles the casting for the locales, but it could be beneficial for the development team to be involved in the locale casting if there are any concerns about the suitability of the different language variants.

3.6 Fonts

Fonts used in the game might be something that are easily taken as granted. But if the game is localized so that it might better reach worldwide audiences, the chances are high that completely different character sets are going to be needed in order to support certain languages. These can include, for example kanji characters used in Japanese and different Chinese language variations, and also Arabic as well as Cyrillic used in Russian.

Sometimes even if the chosen font displays majority of the characters without issues, there can be rarer cases where the font does not handle certain miscellaneous characters. In these cases, the font can display an “error box” symbol in places where it fails to recognize the used character. Mostly if the font displays majority of the character correctly, the error boxes might flag something like line-breaks or apostrophes. This kind of situation is more likely to happen when dealing with switching from Latin to Asian character sets.

But in some cases when dealing with, for example, a newer game engine, Arabic text might prove a challenge. Not only the character set is completely different from Latin, but the language is also written from right to left. This might prove even a bigger issue if the game happens to contain a lot of text, since all of these instances should be adaptable if the writing system is going to change that drastically. This should be given some though as soon as it is known the locale languages include Arabic.

3.7 Localization Quality Assurance

Like all the other parts of the game, every part of the game that has been modified in the localization process needs to be tested. This can be either done internally at the development studio or the process can also be outsourced to external vendors. In some cases, the localization vendor providing the translations might also offer LQA services themselves.

It would be beneficial for the LQA personnel to have access to some kind of debug features in order to speed up the testing process. Any debug tools that let the testers to unlock any kind of collectibles, menus and other instances that contain text throughout the game makes the testing process significantly easier. And more on the VO side of things, a feature that lets the tester to spawn certain NPCs in the various parts of the game world and have them play different voice lines belonging to that character would also save a lot of time. There are many different options what kind of debug features the development team can implement, but the aim would be to make the testing process more flexible for the testers when they do not have to spend time playing the game to access the instances where the texts and VOs are used.

4 Conclusion

The goal of this thesis was to devise a localization pipeline for the customer company that produces AAA video games. The result is clearly defined localization pipeline that covers the whole localization process from start to finish. In a bigger company, like the customer company for this thesis, the responsibility over the localization operations can change relatively often. That is why well documented pipelines are necessary to maintain longevity of any process in the company.

From the customer company's point of view, the authors supervisor at the company feels that what is covered in the thesis is a great communication tool to give people, both inside and outside of the company. It is a good overview of how localization works and what it entails. It is also a great starting point for mapping out craft dependencies in pre-production. In addition, the person who will be taking care of the localization in some of the current game projects feels that the information provided with this thesis will prove useful in the current and future operations.

Then on the other hand, the authors supervisor at the customer company points out that some parts of the localization pipeline could be addressed in much more detail. He points out that the only limitation of the presented work is that it is too high level to allow accurate road mapping and production scheduling. That would require defining separate pipelines for each content content-type to be localized. For example, VO localization from recording of the locale audio to its implementation is something that could be delved much more deeply into. However, the VO localization or even just certain parts of it could be a topic of their own for another thesis. For this thesis where the aim is to define a pipeline and go through all its individual steps, the more generalist approach on the different subjects of the pipeline is much more feasible.

Most of the information and knowhow used to come up with the pipeline sprung mainly form learning things conventionally while the author worked in the customer company. For the author, it has been delightful to discover that the pipeline he assembled shares multiple elements with materials that other industry professionals have come up with as well. The author believes that this, as well as industry insight from his supervisor at the customer company, indicate that the localization processes at least currently are somewhat similar between same sized companies in the industry. That being said, something like indie development studios should be ruled out of the picture when reading this thesis. Due to their small size, some very unconventional practices might spring up, and not only on the field of localization.

One take regarding the whole localization process is that the more money company puts into it, the better the outcome will be. In some aspect that is true, like when talking about the scope of LQA. The more testing rounds there are, the more likely it is that various issues will be identified and addressed. But then if something like the translation quality is being looked at, it is the localization kit that most likely has the biggest impact on the quality of the translations. And that is not necessarily something that can be made better by spending more money to make it. Going further into this, one argument could be that with a bigger budget more proofreading can be arranged and in that way the translation quality can be improved.

Also, localization can sometimes be something of an afterthought for the development team of the game since their main focus is of course on the original game content. This is something that Singh (2011) had also identified in his book "Localization Strategies for Global E-Business". This kind of negligence can be quite detrimental, since the localization process could be made much better if slightly more effort would be put into it earlier. These kinds of things do not necessarily depend on the size or budget of the development studio, but rather if someone is assigned to take an ownership of the localization early enough. It would also be important to make sure that the person put in charge of the localization has the time he needs to think and plan around it.

If the various points of concerns are acknowledged and pre-emptively addressed, the localization process can be a fairly painless undertaking. Everything of course is also dependent on the execution of the various steps. Game development can sometimes, even in the best of hands, be a somewhat chaotic process. But establishing robust pipelines can make a possibility of that smaller.

5 References

Dietz, F. (2007). "How Difficult Can That Be?" - The Work of Computer and Video Game Localization.(5) Retrieved from https://www.raco.cat/index.php/Tradumatica/article/view/75763

Dunne, K. J., & Dunne, E. S. (2011a). *Translation and localization project management*. Philadelphia: John Benjamins Publishing Company. Retrieved from https://ebookcentral.proquest.com/lib/[SITE_ID]/detail.action?docID=786928

Dunne, K. J., & Dunne, E. S. (2011b). *Translation and localization project management : The art of the possible*. Philadelphia: John Benjamins Publishing Company. Retrieved from https://kamezproxy01.kamit.fi:2252/lib/kajaani-ebooks/detail.action?docID=786928

Fernández Costales, A. (2012). Exploring translation strategies in video game localization. *MonTI. Monografías De Traducción E Interpretación,* (4), 385-408. doi:10.6035/monti.2012.4.16

Gambier, Y., & van Doorslaer, L. (2016). *Border crossings*. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Jason Fernando.Pipeline. Retrieved from https://www.investopedia.com/terms/p/pipeline.asp

Labshutz, M., Kr¨osl, K., Aquino, M., Grashäftl, F., Kohl, S., & Preiner, R. (2011). *Content creation for a 3D game with maya and unity 3D*

Mikkelson, N. (2019). Localizing your game: How to make it not suck. Retrieved from https://www.gamasutra.com/blogs/NatalieMikkelson/20190624/345344/Localising_Your_Game_How_To_Make_It_Not_Suck.php

Murauski, A. (2016). 9 tips on localizing audio. Retrieved from https://www.gamasutra.com/blogs/AlexanderMurauski/20160304/267267/9_Tips_on_Localizing_Audio.php

O'Hagan, M., & Mangiron, C. (2013). *Game localization: Translating for the global digital entertainment industry*. Amsterdam: John Benjamins Publishing Company. Retrieved from https://kamezproxy01.kamit.fi:2252/lib/kajaani-ebooks/detail.action?docID=1337555

Singh, N. (2011). *Localization strategies for global E-business*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511920226 Retrieved from https://kamezproxy01.kamit.fi:2252/lib/kajaani-ebooks/detail.action?docID=824462

Stefyn, N. (2019). How video games are made: The game development process. Retrieved from https://www.cgspectrum.com/blog/game-development-process

Timmins, M. (2018). Localization project manager boot camp - part 1 - project kick-off.

Yoccoz, D. (2019). How to create A perfect localization kit. Retrieved from https://gamasutra.com/blogs/DamienYoccoz/20191014/352139/How_To_Create_A_Perfect_Localization_Kit.php