Specification Report

Crowd Sourced Content Aggregation and Suggestion

CS310 Computer Science Project

Project Specification

Jordan Wyatt

Supervisor: Dr. Matthew Leeke

Department of Computer Science
University of Warwick

2015-2016

Contents

This project specification details the proposed improvement and ongoing development to Revolvr, a crowd-sourced media content aggregation and suggestion system developed by Christopher Chamberlain between 2013-2014 [6]. Revolvr is a web based software solution to identifying an individuals *media footprint* via the analysis and processing of homogeneous media items. The suggested improvements include algorithmic focused development, as well as various UI and usability improvements which are outlined in this specification.

1 Introduction

The modern world is vastly connected, with close to half of the population having access to the internet. As of January 2015 there are 3.010 billion active internet users, with 2.078 billion of these users having active social media accounts [15]. These figures are the product of a year-on-year growth, which saw a 21% increase in the number of active internet users, and a 12% increase in the number of active social media accounts in the space of 12 months. With such a rapidly evolving digital landscape also comes an influx in the quantity, accessibility and consumption of online media. However, with such easy access also comes the problem of over saturation. Given such vast quantities of data how can a user identify relevant and desirable content for their tastes?

1.1 Motivation

As discussed in Christopher Chamberlain's previous work on Revolvr, consumers of online media are concious of their *online footprint* [6]. With monumental developments in mobile technology in the past decade mobile and tablet devices now boast a 38% share of worldwide web traffic [15]. As a result, access to the web and its media is easier than ever before and the development of an online footprint is almost unavoidable. The sources of such content vary, from general purpose social networks such as Twitter and Facebook to format specific platforms such as Spotify and Netflix, a user has a wide range of options and discovering content they find relevant often feels like finding a needle in a haystack. Revolvr was developed as an attempt overcome this problem, with its documentation stating 'why settle for what you've already seen, when technology can steer you towards something you may love, perhaps before you know it?' [6].

However, as outlined in the existing project report for Revolvr, limited development time and ensuring the delivered product met the proposed requirements left the software more as a proof of concept, as opposed to an effective content aggregator. The existing work provides a good basis to allow not only improvements to the current algorithm and UI, but also research opportunities to evaluate variations of the implemented algorithm and explore areas of recommender systems, sentiment analysis, and machine learning.

1.2 Project Aims

The overarching aim of this project is to develop a content aggregation and recommendation based on the existing Revolvr framework, with focus on media aggregation, analysis and exploitation of platform specific features, algorithmic development and analysis and user interface design. These aims contain a mix of algorithmic research, scientific analysis, and development / design work which allow the project to have a good depth and breadth. Due to being continuation of work the overhead of setup is reduced allowing focused development points.

1.2.1 Expanded Aggregation

Although an instance of Revolvr has not been deployed on a server yet, inspection of the existing codebase and discussion with the project supervisor led the developer to believe that although Revolvr in its current form can connect to a number of services (Facebook, Twitter, YouTube, Vimeo, Soundcloud and Last.fm) it does not necessarily utilise these services to their full extent. Time will be devoted to ensuring the existing providers are suitable to use in the system, and in this case that the aggregation is effective and distributed between services. This aim will also consider any new services that surfaced since Revolvr's initial development, and aim to incorporate new providers where both possible and suitable.

1.2.2 Exploiting Higher Level Social Media Constructs

The system in its current state only performs aggregation on heterogeneous media source, ignoring meta data and any service by service discrepancies such as followers on Twitter or likes on Soundcloud. Identifying these unique features of each media platform would provide a further link within the system to help develop their media footprint. Examples of metadata that can be used include a users liked Facebook pages, their Soundcloud followers, and who they follow on Twitter. Genre based analysis of songs and videos is also a topic that is open to exploration. This area of improvement hopes to build on the foundations Revolvr has established allowing tailored suggestion for users, whilst at the same time taking an individual services features into account and abstracting these into generalised characteristics for aid in recommendation.

1.2.3 Algorithmic Development

Revolvr currently only uses a single algorithm to make suggestions to users, for the project it is beneficial for the developer both for research and the interests of the product to investigate and implement other recommender algorithms. Allowing a user to choose between a number of different algorithms is likely to provide fresh and perhaps unexpected content to be recommended when their interest is fading. Research will be made into state of the art recommender systems and the approaches they take in their suggestions, and from this a number of bespoke algorithms will be developed for the new iteration of the system. Another area of aggregation that can be improved includes enhanced sentiment analysis. Sentiment analysis is a feature in Revolvr which reduces a piece of medias title to a more meaningful subset of phrases allowing for simple analysis [6, Ch. 5.3.1, p. 40]. However this analytical process does not consider a scenario in which a user shares a piece of media for negative reasons and therefore the associated phrases are negative. Development of this process will aim to analyse the *mood* of a sentence, identifying whether it is positive or negative. The developer also hopes to implement variations of the recommendation algorithm, the results of which will undergo comparative analysis.

1.2.4 Improved User Interface

The original vision for Revolvr was to incorporate jQuery and AJAX to create dynamic pages without the need for a refresh to receive new content, improving the fluidity of the project. The name of the software also comes from the idea of an infinitely scrollable wheel of media which *revolves* [6, Ch. 10.2.4, p. 118]. Implementation of this original design along with the aforementioned algorithmic improvements will create a great deal more of interactivity and immersion. General improvements of the interface will also take place to ensure the application is up to date regarding UI design trends.

1.3 Stakeholders

The main stakeholders that exist within this project are the developer and the project supervisor. The influence and opinion of the previous developer of Revolvr is also invaluable, and their opinion on development will be sought after where possible.

2 Related Work

Consumption of online multimedia is at an all time high, in 2014 the value of the digital music market increased 6.9% to $6.9 billion, representing 46% of global music sales and matching physical sales [12]. Streaming services such as Netflix have also overtaken live viewing this year according to Deloitte's 'Digital Democracy Survey', and content creators on YouTube are earning millions from advertising on their uploaded videos gathering view counts major TV networks dream of achieving [7] [3].

These services, their content providers, and users benefit from algorithms to suggest content to users based on what they like. A platform is only as good as the content it provides, and companies go to great lengths to ensure recommendation algorithms are as accurate as they can be. An example of this was the 2006 Netflix Prize, a *'machine learning and data mining competition for movie rating prediction'* [2]. Knowledge of real life application of recommender systems and the theory underlying it are essential for understanding the existing system and developing it further.

2.1 Recommender Systems

Recommender systems are defined in Resnick and Varian's seminal article as follows: *'In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients. In some cases the primary transformation is in the aggregation; in others the systems value lies in its ability to make good matches between the recommenders and those seeking recommendations.'* [20]. They have proven to be useful means for users to

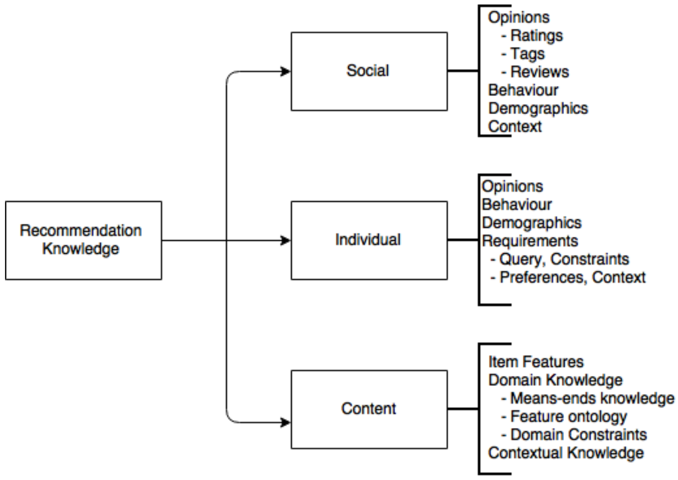

Figure 1: Taxonomy of knowledge sources in recommendation [9]

cope with overload of information and are used extensively in e-commerce. Recommender systems are able to be classified according to the knowledge sources they use, as seen in Figure 1. This taxonomy proposes three types of available knowledge to a recommender system: [9]

- **Collaborative (Social) based knowledge:** knowledge concerning other users.
- **User (Individual) based knowledge:** knowledge of the individual user
- **Content based knowledge:** Knowledge about the items being recommended

A recommender system may use a combination of these knowledge bases, or focus solely on a single one. This taxonomy also helps understand different recommendation techniques.

2.1.1 Recommendation Techniques

As defined in Felfernig and Burke's 2008 paper, a recommendation technique is *'a set of knowledge sources and an algorithmic approach to generating recommendations using those sources'* [9]. The most common recommendation techniques are collaborative recommendation, content-based recommendation and hybrid recommendation. Collaborative filtering is based on gathering information on a user's activities and preferences, and from this predicting what they may like based on similar users. This method benefits from not requiring prior knowledge on an item, so can generally be used without any prerequisite information. In contrast, content-based filtering uses a description of the item and a user's preferences and from this decides if the user will like the suggested item or not. In layman's terms collaborative filtering stems from the idea if two people agree they're likely to agree again in future, and content-based implies a user is likely to seek items similar to what they have liked before. Hybrid recommender systems combine collaborative filtering and content

Table I: Recommendation Techniques

Technique	Background	Input	Process
Collaborative	Ratings from **U** of items in **I**.	Ratings from **u** of items in **I**.	Identify users in **U** similar to **u**, and extrapolate from their ratings of **i**.
Content-based	Features of items in **I**	**u**'s ratings of items in **I**	Generate a classifier that fits **u**'s rating behavior and use it on **i**.
Demographic	Demographic information about **U** and their ratings of items in **I**.	Demographic information about **u**.	Identify users that are demographically similar to **u**, and extrapolate from their ratings of **i**.
Utility-based	Features of items in **I**.	A utility function over items in **I** that describes **u**'s preferences.	Apply the function to the items and determine **i**'s rank.
Knowledge-based	Features of items in **I**. Knowledge of how these items meet a user's needs.	A description of **u**'s needs or interests.	Infer a match between **i** and **u**'s need.

Figure 2: Recommender Techniques [5]

based filtering in a number of possible ways. One approach is to make predictions with each approach separately then combine the results, while in contrast you can design model which unifies the approach of both. Studies have suggested that combining collaborative and content-based filtering and be more effective in a recommender system. Netflix is a successful real world example of hybrid recommender system, collaborative filtering is implemented by comparing watching and searching habits of similar users and recommending content based on a user's previous ratings is content-based filtering.

The techniques discussed above are not exhaustive when talking about types of recommender systems, Robin Burke's 2002paper distinguishes the recommendation techniques shown in Figure 2 [5]. Here **I** is the set of items over which recommendations may be made, **U** is the set of users with known preferences, **u** is the user who requires recommendations, and **i** is some item for which we would like to predict **u**'s preference. Further into the project exploration of the more niche methods will be beneficial, but for specification purposes a high level understanding to plan the project is the key discussion.

2.2 Existing Systems

With social networks and online media platforms showing ever increasing growth the development of recommender systems continues to flourish. To help inspire the development of Revolvr and also see practical examples it is important to consider currently popular and successful applications to see the benefits of these systems. Although the source code and methodologies behind these technologies are mostly private, it is still evident where concepts from recommender systems are being applied.

Social Networks - Social networks such as Twitter and Facebook offer recommendations in the form of suggested friends and followers, their respective names for users of the sites you can connect with. Although the details of how this works are not public, it is likely to be a matrix or graph algorithm which looks at common and different neighbours between nodes to make suggestions. Twitter has also recently implemented a 'While you were away' feature, which aims to *'recap some of the top Tweets you might have missed from accounts you follow'* based on engagement and other factors. Although a small feature in the grand scheme of things, it adds personalisation via recommendation to a user's overall experience [21].

Video Streaming - Online video streaming is arguably the most important main stream commercial use of recommender systems in the past few years. With online subscription based models such as Netflix boasting an impressive 65 million members as of July 2015 [17], it is important from a business perspective to ensure that new users can find content they want, and existing users have such a good experience that they become invested in the product. Netflix suggests movies using a number of advanced machine learning techniques, notably Restricted Boltzman Machines and a form of Matrix Factorization [1] [2]. Although the complexity of these systems are likely to be beyond the scope of this project, their effectiveness is asserted with the fact that *'75% of what people watch is from some sort of recommendation'* on Netflix and the importance of recommender systems within media oriented services becomes apparent [2]. YouTube also offers related and suggested videos based on engagement, promoting videos that have a high average viewing time to video length ratio [27]. These platforms show that a strong recommendation algorithm is crucial when it comes to a dedicated user base and viewer engagement.

Music Streaming - Similarly to the aforementioned video platforms, online music streaming services such as Spotify and Apple Music are often more sought after than physical media. By offering extensive libraries for a low monthly cost and portability via mobile applications they are quickly becoming the preferred method of music consumption for the average user. However with over 30 million songs on some platforms it is important to allow users to discover new content they will enjoy easily so they do not become bored due to lack of new material [24]. Spotify implements both a radio feature and a 'Discover Weekly' playlist. The former discovers songs and creates a shuffled playlist based on the song you choose to deploy the feature on, and the latter is a weekly playlist which provides 20 songs an algorithm believes the user may enjoy based on their listening habits. Spotify use an API provided by The Echo Nest which provides services such as dynamic music data and music discovery and personalisation [16].

These are just a few examples of widely used application incorporating concepts of machine learning and recommender systems at a large scale. Other areas of interest include e-commerce retailers such as Amazon and eBay, who offer both in site recommendations and also external advertisements based on your browsing habits.

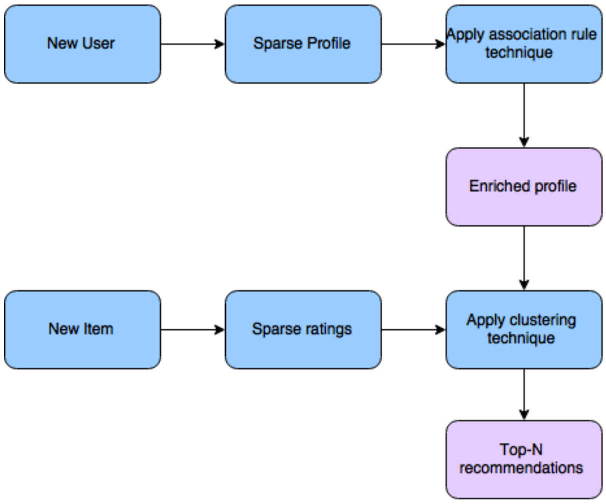

Figure 3: A possible cold-start problem solution [22]

2.3 Cold Start Problem

Recommendation systems often suffer from what is known as the *cold-start* problem. This is defined as when *'recommenders cannot draw inferences for users or items for which it does not have sufficient information'* [22]. This issue can effect both users and items. A new user will be given non personalised recommendations until they have rated enough items to build a profile on their tastes, and a new item in the system will not have any rating information and it is unlikely to be recommended a user. For a user the initial recommendations are crucial on the first impression the system leaves on them, and a poor performance could reduce the appeal of the system.

Two common ways to solve the cold start problem are *association rules* and *clustering techniques*, as discussed in Hridya Sobhanam's 2013 paper [22]. Association rules focus on expanding a user profile so it contains more ratings, and clustering groups of new and existing items based on similarity measures to make predictions for the item. Figure 3 shows one example of a flow of processes that can be performed to attempt to combat this problem. The intricacies of these processes are beyond the technical scope for this specification, but knowing about the problem and the solutions that exist allows peace of mind when designing aggregation systems. Revolvr currently aims to defeat this problem by simply aggregating the most popular content from its services and displaying them to users with no recommendations. Although this isn't fool proof it is sufficient until further work on the problem takes place.

3 Project Requirements

Clear project requirements are vital for tracking progress and providing clear development goals throughout the project. The following requirements outline the current vision for the system for April 2016, but as development occurs are subject to change.

3.1 Functional Requirements

F1: *The system must be able to connect to and retrieve data from a number of third-party sources*

F2: *The system must store and convert data from selected third parties in a common format*

F3: *Data in the system must belong to a group of individuals, or a single individual as a minimum*

F4: *Credentials for the media sources an individual wishes to use data from must be stored*

F5: *Inference must be made to enrich input data obtained from third-parties*

F6: *Multiple models of inference must be available to the user*

F7: *Suggestions should be personalised based on an individual's tastes*

F8: *The system should be able to group users, and provide suggestions based on the preferences of these similar grouped users*

F9: *The system should be able to recommend to new users whilst avoiding the cold-start problem*

F10: *User feedback must be fed into the system to refine the inference mechanism*

F11: *The media must be consumable within the application by the user, without deducing from the experience provided by the original source platform*

F12: *The inference mechanism must exploit platform specific features to enhance the recommendation experience*

F13: *The system must provide an 'infinite' revolving wheel of recommendations on an easily accessible home screen*

3.2 Non-functional Requirements

NF1: *The system must be modular, to allow extensibility*

NF2: *The system should be maintainable, commented and provide strong documentation for future work*

NF3: *The system should provide a seamless user experience, more so than the original implementation*

NF4: *The system should be designed so as to allow scalability*

NF5: *The system should be preferable for discovery as opposed to using the individual services separately*

3.3 Hardware and Software Constraints

As with any software application optimisation in regards to scalability is an unpredictable factor. For the whole duration of this project it is likely the server will be on a cheap, low capacity hosted server due to available resources. As a result this means ensuring the software can cope with extreme amounts of data and users will be a challenge. It must also be ensured that any additions to the existing system aim to impact performance positively and do not reduce the quality of the existing software solutions. The new revolving content wheel may be a challenge to incorporate on mobile devices, and in the case that it isn't feasible an alternative option per device must be developed.

3.4 Foreseeable Challenges

The large existing code base is yet to be deployed and explored, changes to the above requirements and the details listed in this specification are subject to change based on the pace of the project and unforeseen circumstances.

4 Legal, Ethical, Social and Professional Issues

As with all software products, especially ones of such a personal nature, it is important to ensure the product follows standards and can be trusted by users. This section of the specification will discuss the possible issues that could arise during the development of Revolvr in the legal, ethical, social and professional domains without careful planning and consideration. To assist the developer the British Computing Society Code of Practise was taken into account, the issues discussed are not exhaustive but instead are the most applicable [23]. The previous developer also ensured the work that was left adhered to these standards [6].

4.1 Legal Issues

Due to the nature of the system, Revolvr will be processing and displaying media from third party sources. A clear concern here is the topic of intellectual property and licensing. In regards to licensing, Revolvr will only be aggregating content from trusted and established sources that themselves adhere to the correct licensing laws and terms and conditions. As a result of this the system must also comply to the same standards that the third party providers do. Intellectual property will be preserved in a number of ways, firstly the API's for the aggregated services all provide watermarks on the embedded players to show the source of the content, and by following the link these watermarks provided the original uploader receives credit too. A software license agreement will also be developed which state where the products obligations lie.

Although legal issues are an important topic in all products, in this case the developer and the project supervisor are confident that by using trusted and reputable aggregation sources such problems can be avoided.

4.2 Ethical Issues

The key ethical concerns with user oriented sites is data privacy, and with media oriented products also comes the task of filtering explicit content. The former will be guaranteed by encrypting any sensitive user data such as passwords, and Revolvr's philosophy is concerned with respecting the user so their data will not be passed to third parties in any circumstances. Regarding suggested content of explicit nature the platforms Revolvr uses will themselves provide filtering of explicit content (i.e YouTube's age gate feature). This obviously is not fool proof, however this is again a situation where the system can rely on the third party content providers to do their jobs by being reliable and trusted platforms. Although this may come across as a lack of planning or carelessness, Revolvr will ensure it only aggregates from sources who the stakeholders believe fit the mould in regards to the provided content the provide and its suitability for the system.

4.3 Social Issues

Social issues are often the ones that can make the user feel victimised, as opposed to approaching from a business or financial perspectie. Revolvr will thrive for inclusivity, ensuring that perspectives such as cultural, social, gender and disability are taken into account into the refreshed design. This will be inforced by ensuring the technology is available to all, and does not cater to a specific race or gender. In terms of disability the system will take into account colour blind and disabled users, by ensuring multiple queues are available to promote the functions of the product and making the UI clear and intuitive. Although multiple languages are likely to not be supported due to time constraints and lack of resources the design of the UI will be approached such that both visual and language prompts are provided.

4.4 Professional Issues

Being a user oriented service it is important to value the user to create a sense of worth for the platform and a relationship of trust. As previously mentioned all data provided by a user will be kept private and only used for the service they receive. The previous developer also ensured all services included used secure channels and session tokens, this ensures that the connections to the providers cannot be exploited by malicious third parties if an attack were to occur. Finally, the developer will act according to the aforementioned BCS code of conduct by showing public interest, professional competence and integrity, duty to relevant authority and showing duty to the profession [23].

5 Project Management

Due to this project being a continuation of existing work, there is a significant amount of overhead with regards to understanding the product and the concepts it utilises before moving forward. As a result it is important to have a solid plan to ensure work continues to be efficient, whilst still allowing time for contingencies and set-up periods. However, situations can occur with plan driven development where deadlines can act as contingencies themselves and therefore flexibility will be provided by using an agile approach which provides lenience in proposals contained in this specification.

5.1 Design Approach

An agile approach will be taken during the development of this project, both in terms of the software development itself and also the project documentation and management. This allows structure and sensible deadlines, whilst still providing the ability to respond to unforeseen circumstances. The software methodology intricacies are further discussed in section 5.3 of this specification.
A weekly informal Wordpress blog will be updated weekly to map progress for the developer and supervisor, discussing the prior weeks achievements and the coming weeks aims to ensure a record of work is kept [26]. Alongside this the report will be compiled throughout the whole project to ensure the latest developments can be explained in good detail without long term memory recall. Finally, Git will also be used as a version control with appropriate commit messages to ensure the software itself is organised. This design approach ultimately aims to provide a structured and focused development process, whilst still allowing experimentation and change if deemed necessary or beneficial for the project.

5.2 Project Timeline

The project will aim to follow the deadlines proposed in the Gantt chart showing in Figure 4 and the accompanying table of dates in Table 1. Consideration for refinements and user feedback are taken into account towards the end of the project, during this period the developer hopes to be applying tweaks with the core functionality and additions tested and functioning correctly. This Gannt chart at this early point in the project can be subject to change due to the agile nature discussed previously, and will be revisited in the forthcoming progress report.

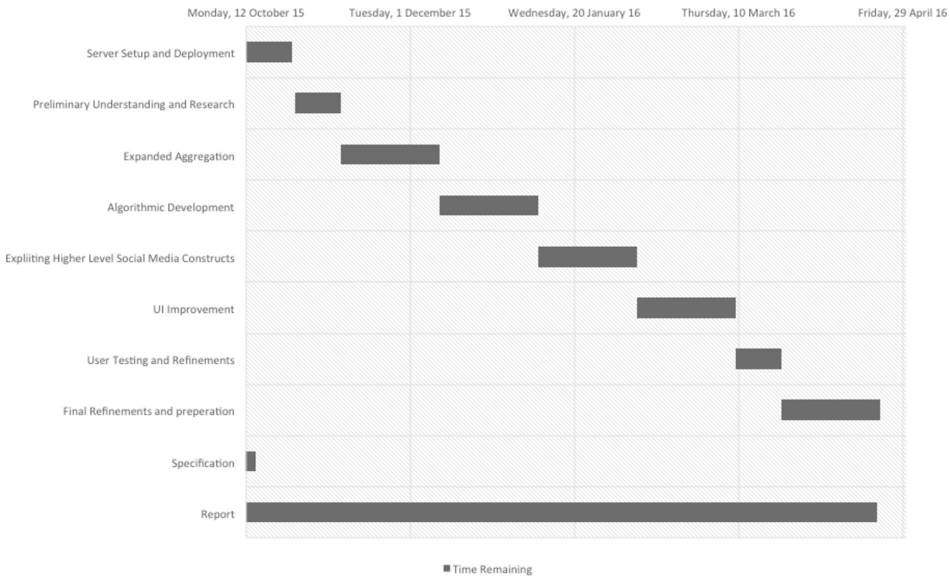

Figure 4: Gannt chart showing proposed project deadlines

Table 1: Proposed Deadlines

Process	**Proposed Start**	**Proposed End**
Server Setup and Deployment	12 October 2015	27 October 2015
Preliminary Understanding Research	27 October 2015	10 November 2015
Expanded Aggregation	10 November 2015	10 December 2015
Algorithmic Development	10 December 2015	9 January 2016
Exploiting Higher Level Social Media Constructs	9 January 2016	8 February 2016
UI Improvement	8 February 2016	9 March 2016
User Testing and Refinements	9 March 2016	23 March 2016
Final Refinements and Preperation	23 March 2016	22 April 2016
Specification	7 October 2015	15 October 2015
Report	12 October 2015	30 April 2016

5.3 Software Development Methodology

Similarly to the design ethos discussed in Section 5.1, flexibility with focus is the key to this project and therefore a hybrid agile development methodology will be undertaken. A concrete classification such as SCRUM or Extreme Programming will not be beneficial to this project as the developer is the only person working on the project and most agile methods are defined for a team. Instead, a flexible gannt chart subject to change where necessary (see Section 5.2) has been constructed, where each time period can be seen as a sprint with set yet open ended design goals.

5.4 Tools

Revolvr's existing documentation has provided a comprehensive list of tools used in its development [6]. At the current point in time the developer believes this list will remain mostly the same and the key aspects are given below, however if a proposed aim requires the addition of a new technology this can be adopted due to an agile approach and a modular design. Further details on this will be provided in the progress report when the developer has fully hosted and analysed the existing product.

5.4.1 Data Collection and Analysis

Python will be used to collect and analyse the data, this choice was made due to the wide variety of external plugins and modules available for working with data as well as the external platforms the system exploits [19].

5.4.2 Data Storage

The existing solution is built using MongoDB, an open source, document-oriented database. Mongo focuses on both scalability and development agility and therefore allows quicker work than conventional database solutions such as MySQL[13].

5.4.3 Data Visualisation

A combination of Ruby on Rails and Bootstrap will be used to develop the front end [11] [4]. Ruby on Rails will provide the content for the user interface, and Bootstrap will be used to provide sleek and consistent styling which performs well on both conventional and mobile devices.

5.4.4 Organisation and Communication

Organisation is key in a project of this size, and therefore exploiting tools to help direct and maintain the project is extremely beneficial. As mentioned previously a Wordpress blog has been created to track progress, to accompany this a Dropbox folder is being shared with the project supervisor to monitor all written work as well as code [25] [8]. Version control will rely on Git using a private GitHub repository to protect intellectual property, and all other communication will occur in person, email or instant messaging [10].

5.4.5 Other Considerations

As shown in Figure 4, the setup period and learning the required technologies is an ongoing process so implementations of the web server and smaller details are currently undecided. As of the time of writing, an Ubuntu server provided over the cloud by DigitalOcean is being setup to host the application on which is running nginx [18]. The server has 512MB memory with 20GB disk capacity, and is hosted via a SSD cloud server for optimal performance. The previous iteration ran on Apache but migrating to nginx should not pose any significant issues. Further details on this will be provided in later documents when the existing code base has been deployed successfully.

6 Testing and Success Measurement

Comprehensive testing using a number of established strategies will be used to ensure the project meets the specified requirements. The organisation and quality of the codebase will also be maintained throughout the project.

6.1 Testing Strategy

The testing strategies implemented are outlined below which include unit, integration, system and user testing.

6.1.1 Unit Testing

Unit testing is focused on verifying the functionality of a specific section of code such as individual functions or classes. All original and new code will have multiple tests to catch issues such as branching errors or corner cases. The relevant testing framework for the incorporated technologies being tested will be learnt and applied to reduce uncertainty.

6.1.2 Integration Testing

Once unit testing provides sufficient results, the tested modules will be integrated with one another to test functionality in a larger scale, moving closer to the real world execution conditions. This area of testing will aim to expose defects in the interaction between components as opposed to errors with the components themselves.

6.1.3 System Testing

Following integration testing system testing will be thoroughly applied, which involves testing the system as the whole software solution that the end users would interact with. This will ultimately be the testing stage which verifies the proposed requirements in Section 3 are achieved. However, the prior testing stages cannot be neglected as optimal results in unit and module level tests are necessary to ensure system testing is ready to occur.

6.1.4 User Testing

Live user testing will be invaluable to the final system due to being such a user oriented service. Once initial developer testing has been signed off peers of the developer and the supervisor will be given access to an alpha, hopefully providing useful feedback on the functionality and quality of the system.

6.2 Success Measurement

A lot of questionable metrics are suggested for measuring success within a software engineering project, a task which becomes even more of a challenge when implementing a nondeterministic agile development methodology. This project will measure the success of the project based on the concepts delivered in The University of Warwick's CS261 Software Engineering module [14], which are the following:

- How does the software meet the original specification?
- How does the software meet the customer's expections?

Quantitative analysis will take place in the form of analysing the number of functional and non-functional requirements that are achieved, which will explore whether the project fulfils its original specification. The customer's expectations are qualitative, subjective metrics and although difficult to draw conclusions from will be taken into account by providing questionnaires or surveys to the users testing the software.

7 Conclusion

To summarise this specification, the project aims to build develop a system for content aggregation and recommendation based on the existing Revolvr framework. The emphasis on development will be focused on the aims outlined in Section 3 of this document, specifically these areas are media aggregation, analysis and exploitation of platform specific features, algorithmic development and analysis, and user interface design. By utilising the promising foundations of Revolvr, a project compromising of both a development and research element can be undertaken with the final goal of presenting a system which can not only assist the user, but take them on a personalised journey of content discovery

> ‘A lot of times, people don't know what they want until you show it to them."
>
> Steve Jobs

References

[1] Xavier Amatriain. How does the netflix movie recommendation algorithm work? `https://www.quora.com/How-does-the-Netflix-movie-recommendation-algorithm-work`, 2015.

[2] Xavier Amatriain and Justin Basilico. Netflix recommendations: Beyond the 5 stars (part 1). `http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html`, 2012.

[3] Damon Beres. Youtube stars' huge earnings will make you question your life choices. `http://www.huffingtonpost.com/2015/02/05/youtube-stars-money_n_6549906.html`, 2015.

[4] Bootstrap. `http://www.getbootstrap.com`, 2015.

[5] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[6] Chris Chamberlain. Revolvr: A crowd-sourced media content aggregation and suggestion system, 2014.

[7] Deloitte. Digital democracy survey: A multi-generational view of consumer technology, media and telecom trends, 2015.

[8] Dropbox. `http://www.dropbox.com`, 2015.

[9] A. Felfernig and R. Burke. Constraint-based recommender systems: Technologies and research issues. In *Proceedings of the 10th International Conference on Electronic Commerce*, ICEC '08, pages 3:1–3:10, New York, NY, USA, 2008. ACM.

[10] GitHub. `http://www.github.com`, 2015.

[11] David Heinemeier Hansson. `http://www.rubyonrails.org`, 2015.

[12] IFPI. Ifpi digital music report 2015: Charting the path to sustainable growth, 2015.

[13] MongoDB Inc. `http://www.mongodb.com`, 2015.

[14] Stephen Jarvis. Cs261 software engineering - project management. `https://www2.warwick.ac.uk/fac/sci/dcs/teaching/material/cs261/SE-L8.pdf`, 2015.

[15] Simon Kemp. Digital, social & mobile, 2015.

[16] The Echo Nest. `https://press.spotify.com/us/information/`.

[17] Netflix. Q2 15 letter to shareholders. `http://ir.netflix.com/common/download/download.cfm?companyid=NFLX&fileid=839404&filekey=C3CE9EE2-C8F3-40A1-AC9A-FFE0AFA20B21&filename=FINAL_Q2_15_Letter_to_Shareholders_With_Tables_.pdf`, 2015.

[18] Digital Ocean. `http://www.digitalocean.com`.

[19] Python. `http://www.python.org`, 2015.

[20] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[21] Paul Rosania. While you were away... `https://blog.twitter.com/2015/while-you-were-away-0`, 2015.

[22] Hridya Sobhanam and AK Mariappan. Addressing cold start problem in recommender systems using association rules and clustering technique. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, pages 1–5. IEEE, 2013.

[23] British Computing Society. Ethics of computing. chapter British Computer Society Code of Practice, pages 102–110. Chapman & Hall, Ltd., London, UK, UK, 1996.

[24] Spotify. Information. `https://press.spotify.com/us/information/`, 2015.

[25] WordPress. `http://www.wordpress.com`, 2015.

[26] Jordan Wyatt. Cs310 blog. `www.cs310blog.wordpress.com`.

[27] Youtube. Changes to related and recommended videos. `http://youtubecreator.blogspot.co.uk/2012/03/changes-to-related-and-recommended.html`, 2012.