Predictive Analytics for New Product Development

Student: Josh Hills 140177712
Course: Computing Science Bsc Honours w/ Industrial Placement (G401)
Date: 4th May 2018

Supervisor: Dr. Matthew Forshaw

Words: 14,955

Newcastle University

1. Preamble

1.1 Abstract

This paper covers the construction of a data-mining architecture sufficient to explore the problem space of the new product development lifecycle of software projects using engagement data, and furthermore quantifies to a degree the relationships between metrics during this time period.

Research is conducted towards the heuristics of agile project management, and furthermore the practical application of business strategy in the context of the video games industry, which is the primary archetype throughout. Additionally, the scientific principles used to explore the data are covered.

Through statistical analysis it is outlined that the data collected exhibits bias due to the context of its submission, and can be used to both classify the type of 'story' a specific product has, and tune process of active development to maximise the success criteria of a project.

The insights derived from analysis are placed into an industrial context through an evaluative survey, and intended to be used as a catalyst to provide teams undertaking rapid prototyping with justification and context for their decisions, with further application in predictive modelling.

1.2 Declaration

I declare that this paper represents my own work except where otherwise stated. It is in accordance with University and School guidance on good academic conduct.

1.3 Acknowledgements

I would like to thank my supervisor Dr. Matthew Forshaw for his continued support throughout this project.

I would also like to thank Joel Graham and Erik Lagel of Jagex Games for acting as valuable industry sponsors, in addition to a variety of other professionals and friends who took the time to provide feedback along the way.

1.4 Contents

2 Introduction

2.1 Motivation & Rationale

I have undertaken this project after working in the games industry and noting that many problems with new ventures centred around a lack of explicit direction at a project-management level.

Although video games are distinct to traditional software products, their process of development shares many traits and interchangeable terminology, and it is not unreasonable to suggest that the outcomes of this project may be cross-applied.

2.2 Context

Data Backing New Ventures

Companies are increasingly reliant on the existence of analytical data in risk mitigation for their ventures. This is seen in comprehensive services now exist to augment critical business decisions and integrate the product-lifecycle-management process into a scientific context[i]. Given the right conditions these systems can predict trends and suggest appropriate changes to help ensure project success.

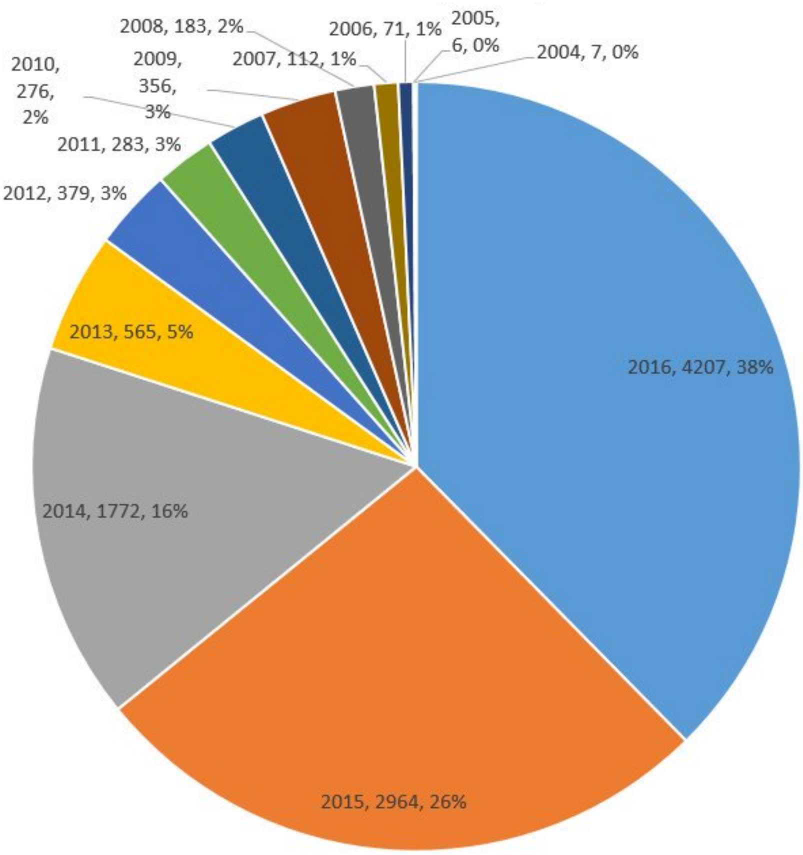
Most companies allocate resource to the prototyping and validation of new ideas in greenfield development. If the teams that are responsible for this have constrained resources or limited active and historical feedback, there may be an inability to derive meaningful conclusions from data collected. The resulting methodologies employed during research and development (R&D) are therefore often heuristic, which can be a hindrance to rapid innovation[ii].

During new product development (NPD), although many decisions must be made recurrently, most people view products atomically; few consider the unique evolutionary nature of software, which is becoming intrinsic to securing a return on investment (ROI). More focus may be placed on planning the product prior to release, as there are many unknowns post-release.

While there have been many studies regarding the prediction of product success in the intersection of business and data science, few have explicitly studied the effects of this process of iteration. This is in part due to the relatively recent emergence of mass distribution of digital goods and data pertaining to them, and the resurgence of popularity in machine learning applications (MLA)[iii]. This is also a result of the marked variance in business strategy from organisation to organisation.

Games Industry

The PC games industry provides an apt archetype for this problem, as studios, often lacking time and money, can be entirely reliant on the success of new products. While distributors are under increasing social pressure to provide transparency in their transactions[iv], the sheer quantity of observable work is skyrocketing in tandem with their success.



Number of products published to Steam by year[v].

Prototyping Methodologies & Release Models

There are a number of ways in which a software project may be developed and delivered upon. Each categorisation is further transmuted by the individuals working on it.

Since 2012, a release model of ‘early access’ has arisen whereby games are made publicly accessible before they are considered ‘complete’, and undergo a reactive metamorphosis for a period of time until they are considered so. Although it is unclear whether any one process is better than another, it has blurred the lines of historically standard publication[vi]; what the product represents and provides can be extended but also replaced. Furthermore, many metrics attached to this process can now be publicly scrutinized. Some metrics (e.g. ‘sentiment’ towards current content) are crowd sourced, and can prove influential in informing companies’ decisions; most notably as to when to ‘fail’ or transition out of early access. By shirking more stoic release models of the past, and adopting ‘agile’ methods, game developers can tailor their content at each update to maximise the potential of eventual accumulative success, similar to how application developers introduce new features to degrees of public reception.

The uncertain nature of this work is a common grievance among developers that I have encountered during my professional work within the industry, and a balance must be sought between creative liberty and economic viability.

2.3 Aim & Objectives

Aim

To use insights derived from data-mining product analytics to improve decision-making in the new-product-development lifecycle of software.

Functional Objectives

FO1. Survey the use of analytics and machine learning in digital product development lifecycles.

In order to leverage existing academic material via a literature review to ground the project in context and identify avenues of exploration.

FO2. Create a system architecture capable of aggregating historical data in the problem space.

In order to process all relevant information from every ‘early access’ product within the timeframe set forth by the project plan.

FO3. To quantify the correlation of product updates with trends in user sentiment and engagement using statistical analysis.

In order to supplement developers with accurate insights into the impact of their decisions, and thus help them tailor their release strategy to maximise their success criteria.

FO4. To assess the integrity of the findings in a real-world business context.

In order to meaningfully evaluate the success of the project and its applicability to the original problem.

FO5. To develop a public interface to the data share findings.

In order to promote its usefulness in cross-application to other software with similar available data using transparent documentation, and furthermore reproducibility.

2.4 Alterations

The project proposal contained elements of machine learning on the assumption that this would be necessary in order to take advantage of trends and furthermore automate the process of analysis for future products. It was decided in communication with an industry sponsor that since the problem space has an overwhelming number of externally influencing factors ('large dynamic range'), that it would be more beneficial to focus explicitly on asserting assumptions through more standard means[2.4]. Ergo, the elements of 'prediction' manifest as the generalisation of findings into bootstrapped advice for developers, and further suggestions for their application in a predictive manner.

2.5 Methodology

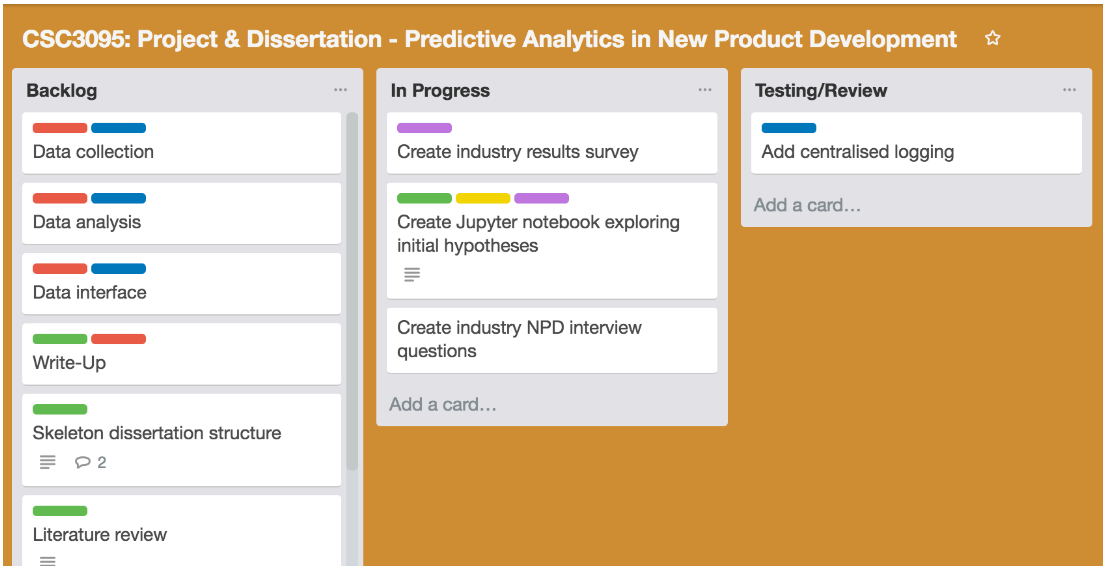
High-level Overview

This section doubles as a descriptor of the structure of the rest of the paper. Following industry guidance, the steps necessary to complete the project were agreed upon:

- Research
- Data collection
- Data processing (cleaning and manipulation)
- Data analysis (exploration and visualisation)
- Evaluation

The first significant chunk of practical implementation entailed building a system to collect and process data. The second significant chunk of work was the analysis of the data collected.

Project Management



Examples of work tickets in 'Trello'.

Work was undertaken with the agile software development methodology Kanban[vii]. This suited the adaptive nature of analysis, and allowed the objectives to pivot on the affordance or lack thereof of investigative leads presented by the dataset.

Analysis

An approach to statistical analysis was also formalised to aid project planning and documentation. Each iteration built upon the findings in the previous step.

SECTION	EXPLANATION

IDENTIFIER	For reference.
AIM	Hypothesis or specific feature to investigate.
OPERATING ASSUMPTIONS	Accumulative hypothesis verified in some way.
PURPOSE	The onus for this being explored as the next step.
INVESTIGATION	Explanation of the problem, manipulation and visualisation of data, with statistical functions observed. Repeated *n* times until aim asserted.
RESULTS	Insights summarised.
POTENTIAL APPLICATION	Suggestions to utilise insights (if applicable).

2.6 Definition of Terms

Data set	When mentioned generally, this refers to the data set collected specifically for use in this paper.
Video Game	Though many subjective definitions exist, for the context of this project this is defined generally as a creative software product.
Viral/virality	The quality of something having been circulated rapidly and widely.
Developer	Parties that develop software products.
Publisher	Companies that pay commissions for rights to publish products.
Distributor	Intermediary role between publisher and retailers.
DLC	Downloadable content that extends a product.
Predictive	The quality of predicting an event or result.
Analytics	Information resulting from the systematic analysis of data or statistics.
Greenlighting	The decision to begin work on something.
User score	The number denoting the community perception of a product on a scale of 0 - 100, as a ratio of positive to negative reviews.
Churn	The notion that something will diminish over time, specifically players of video games.
ETL	Extract, Transform and Load is a process of moving data between systems.
EDA	Exploratory Data Analysis is a process in which data is explored for use.
Changepoint	Point of aberration in a stream of information.
Hyperparameter	Modifiable setting that precedes an operation.
Preprocessing	The process of altering raw data as it enters the system.
Munging	Also ‘wrangling’, the process of transforming data for use.
CCU	Concurrent Users at a point in time.
Changelog	A list of content updates to a product.
Greenfield	Pertains to unexplored ideas
API	Application Programming Interface exposing data from a running program, usually through web protocols.

3 Background

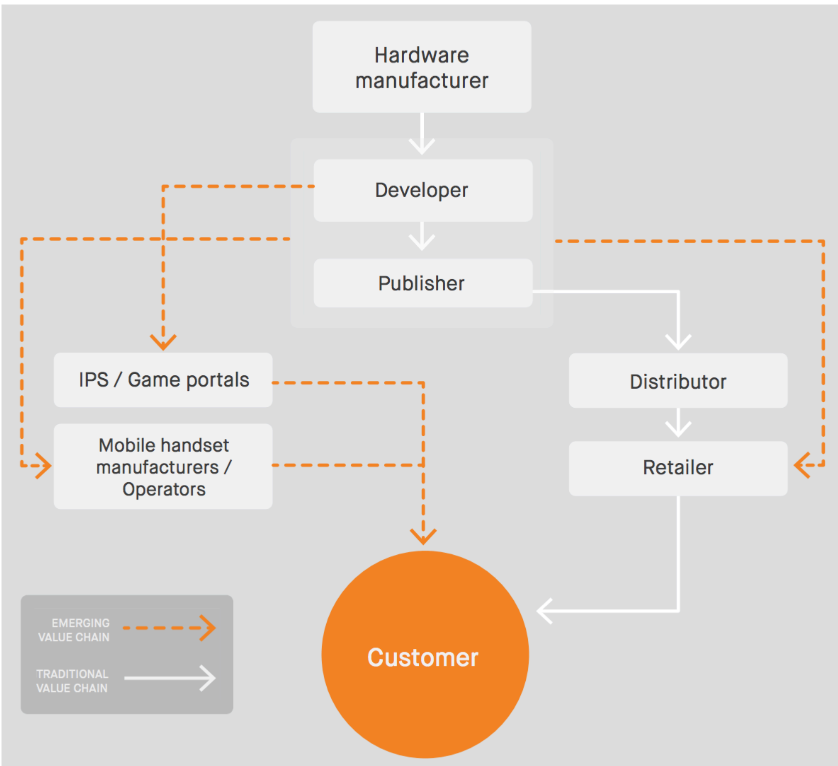
3.1 Theory

3.1.1 New Product Development

Aim

By understanding how decisions are made for new ventures, it can be better understood what is desirable and problematic. This research therefore provides a justification and focus for the project, as it highlights many promising areas of exploitation while simultaneously describing a slew of uncertainties that contribute to their volatility.

This paper foregoes many business concepts pertaining to corporate strategy to focus on the practicality of active design, development and marketing as a mixture of achieving project deliverance. Nonetheless, there is an inescapable cascade of considerations present in the process.



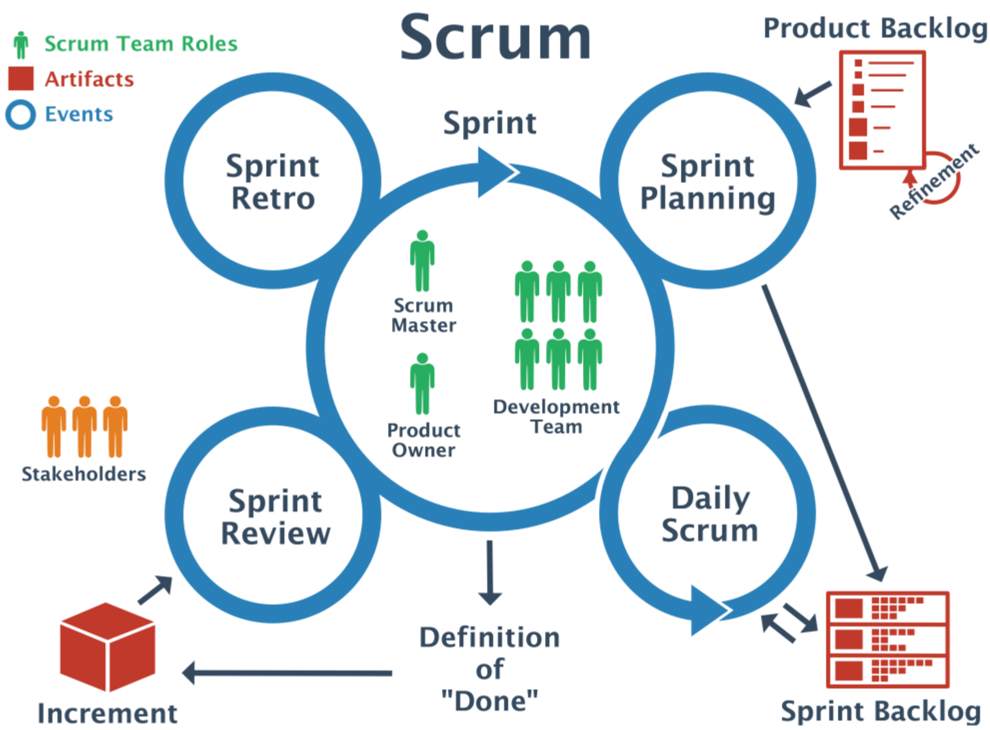
The emerging video-game value chain[viii].

The step that presupposes this research is the decision to develop a software product. Sometimes referred to as 'filtration', it demands questions of an organisation as a whole: whether a product is a good 'fit' for a company based on current products, employees, fan-base, the market etc. On the assumption that this has already been completed and a specific product has been 'greenlit', research is henceforth dedicated to the area this paper aims to benefit – active creation of new products in a state of 'early access'.

Development Models

Prior to publication, the process of completing work must be addressed. Software development has a storied history of practical change. Most obvious is the shift from 'waterfall' models towards 'agile' development, a set of principles that theoretically enable more responsive development. This has been driven by market demand for better quality games in shorter timespans[ix].

In this process, teams 'inspect and adapt' while stepping towards milestones, with the retrospection of 'Scrum', an implementation of 'agile', purportedly improving the development of video games[x] across the industry and creating a "a framework for close collaboration and iteration"[xi].



Jordan Job's summary of Scrum[xii].

Embedded in this iteration is the need for evaluation at a specific point in time in order to inform future decisions. Although there are abundant resources pertaining to assessing the performance of individuals co-ordinating to deliver new features, it is not often considered that the success of those features is a determinant for future ones. It may be the case that an undesirable outcome is reached, such as the closure of an avenue or entire project under the 'fail-fast' mantra[xiii].

The almost vulnerable nature of 'early access' disrupts 'agile'[xiv] even as it is adopted rigidly. This dependency on 'externalities' is described as a stopping force by *x*, as the uncertainty of "every potential endgame on its expectations about events that will happen or on the behavior of other players", requires that "innovators… defer decisions until they have more information about the innovation's fate and other players' experiences with it"[xv].

It is abundantly clear that due to changing processes within an organisation, that being able to understand and predict the impact of decisions throughout development is a prerequisite for taking risks: "by analysing historical data, results can be measured, key lessons learned, and better initiatives employed"[xvi].

Crowd-Sourcing

Technological items on crowd-funding sites receive investment from users who by doing so presumably accept the pitch in its current form, and typically then wait for it to be 'finished'. This does not preclude feedback, however in order to secure investment a strong creative vision must be purported from the offset. This is supported by studies show that the more comprehensive the initial proposal appears, the more chance it has of succeeding[xvii]. User agency appears to manifest in engaging with each other. Although this shares similarities with the 'early access' model, the latter differs due to the omnipresent affordance of immediate interaction with the product.

Historically, there have been various methods by which companies have involved potential users in order to dispel cognitive bias during development, and cater to a specific type of audience. It is shown that games that cater to a niche at a specific point in time (where there are a lack of other similar products) perform better due to a lack of competition[xviii]. Furthermore, by monitoring the habits of users, different cohorts (such as 'whales') have been identified that show an imbalance in return-on-investment; 50% of the mobile gaming market revenue was generated from 0.15% of users[xix]. Therefore, the notion of responding to feedback resonates with studies that show that a strong relationship with a minority of fans can yield the most positive outcome.

This classification of users is further employed in the process of attaining feedback, where companies continue to invite those most influential to 'alphas' and 'betas', allowing users to interact with a product before it is considered finished. This form of play-testing, sometimes conducted on-premises, has been important enough to have been legitimized into a growing number of employment opportunities[xx], and refinement through user testing is described as necessary, as "interaction… is something too subtle and too complex to script out in advance, requiring the improvisational balancing that only testing and prototyping can provide"[xxi]. Further research has enhanced the role of users, studying co-development through virtual interaction design dubbed 'Community Based Innovation'[xxii] (CBI). These principles have been elaborated upon to incorporate user satisfaction into psychometric models to aid evaluation[xxiii], and have seen great success in a theoretical setting.

A user play-tests a game in a controlled environment[xxiv].

How, when and to what extend to crowd-source feedback during new product development has been the subject of many theoretical studies. Advancements in digital distribution now “provides a quasi-direct relationship of the development companies with the end users”[xxv].

It is therefore apparent that user feedback is vital to the development of new products. To be able to screen content prior to an official, wider release against an accumulative number of invested parties is not unique to the ‘early access’ model, however there now exists a platform for polling ‘big’ data, and this represents the possibility of a hitherto unseen level of investigation.

<u>Defining Success</u>

Success for new software products can manifest itself in server ways. Most are transient to meeting or exceeding fiscal targets, however this is composed of a combination of other desirable properties quantified as ‘Key Performance Indicators’ (KPIs). After analysing the meaning of ‘value’ in a business context, successful innovation is defined by as the “process of turning ideas into valuable ideas, with potential of generating sustainable benefit for the organization seeking to monetize this added value”. This encompasses measurements of use, improvement and satisfaction[xxvi], and reflects the increasing number of stakeholders involved.

As we have seen, in some cases success is promoted via ‘hyper-personalisation’ of content as a means to foster a relationship with consumers. These methods are drivers for further direction, and represent a “service mentality that clearly is the next logical step in the evolution of franchising and serialisation”[xxvii]. It was found that by compounding the measurement of user involvement with more discrete metrics, scores could be created representing the extent for which a game achieved its goals during each iteration[xxviii]. This is mimicked in more theoretical studies that attempt to synthesize success in an abstract, advisory fashion; one defines phases in a conceptual model, at foundational and project level, recommending "early customer involvement" and the "screening of [a] preliminary product concept" and "refinement" as essential to performance[xxix].

This is exemplified in-industry, where "data informs creativity"[xxx] to the extent that the current success of a product determines its own journey, with case studies of DLC for video games serving as examples of developers leveraging already popular-in-game content for further ideation. In this sense, while Schwaber's 'lexicon of done'[xxxi] considers iteration towards a state of completion, it fails to address the cyclical nature of 'completion' itself.

It is therefore apparent that success is driven by a symbiosis of what is desirable by organisations and users of the product, and to be able to measure both of these as early as possible is vital to coordinating the NPD lifecycle.

Transitional State

At some point if a product is successful in achieving its aims, it should transition from its 'pre-released' state to a one that is 'released'. Although this is often accompanied by an effort to disassociate itself from its previous state, the disparity of these states may be minor in regards to the content of the product. Ironically, the ramifications can be huge depending on how the intentions of the company have been communicated to users – e.g. if a product, once released, will no longer receive updates, this may raise concerns.

This communication is intrinsically linked to marketing. It is argued that the landscape of marketing software products has changed fundamentally; it now must be "deeply embedded into actual product development", as "to succeed with any meaningful degree of effectiveness, it must also serve as a trusted and transparent vehicle through which the user ultimately feels he or she achieves some degree of participation in (and influence over) the shape of the end-product"[xxxii]. The need for this new type of slower-burning 'discovery' indicates that throughout 'early access', positive engagement is nurtured, and must hail from a mixture of sources recurrently.

It is therefore evident that developers must create a continuous feedback loop in order to transition to a state of success effectively, regardless of the body of work. This is something supported by the platform of study.

Summary

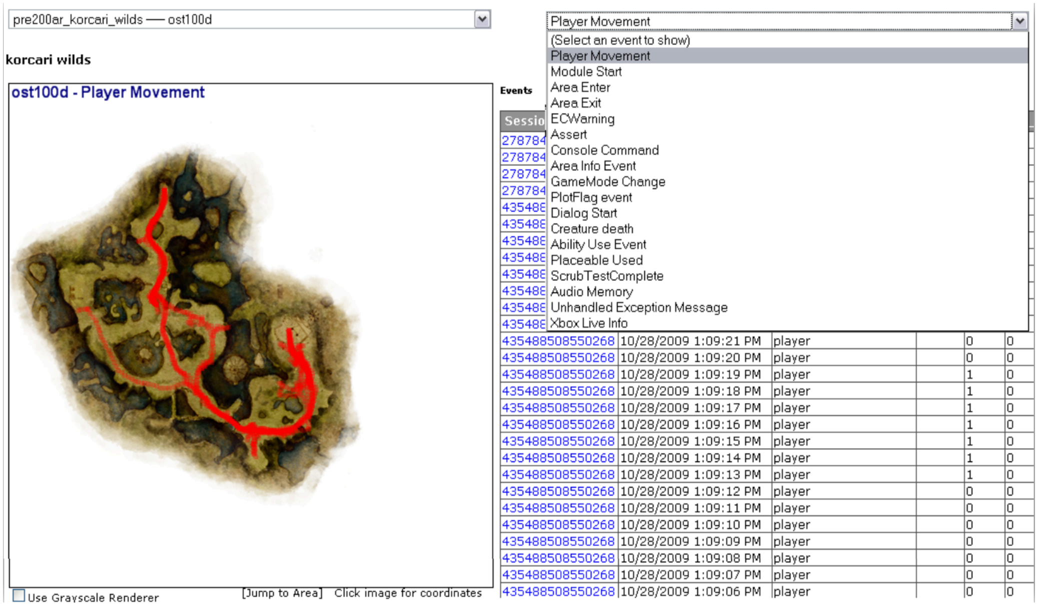
A failure of most sources is simply a product of the fast-paced nature of game development. Most studies do not prove long-term effectiveness of their results, nor do they consider that 'early access' makes publication iterative. They provide a wealth of advice and opinion in the process that is not employed at a large scale, with Füller stating that "the actual market impacts of co-developed products could not be tested on a large scale as most of the CBI projects are not carried to the point of testing the innovations in an actual marketplace"[xxxiii]. This could be related to how developers share information.

Therefore, this paper takes advantage of prior findings and applies them to this updated context, attempting to collect information about and find patterns in the same areas of importance against the benchmark of a larger data set. This is reflected in the metrics chosen, which encompass the areas of developer communication in the form of updates, community feedback/sentiment in the form of reviews and success criteria in the form of usage.

3.1.2 Games Analytics

Aim

The aim in researching games analytics was to understand the kind of procedures already being applied to information in well-defined systems to benefit the development of new products. It was not to recant case studies, but rather to collect the general motivation for researching integrating predictive analytics into a workflow. The technicality of games analytics is not dissimilar to regular analytics – ‘hooks’ into products deliver information to stores for investigation.



A telemetry tool provides automatic visualisation of in-game events[xxxiv].

Information Economy

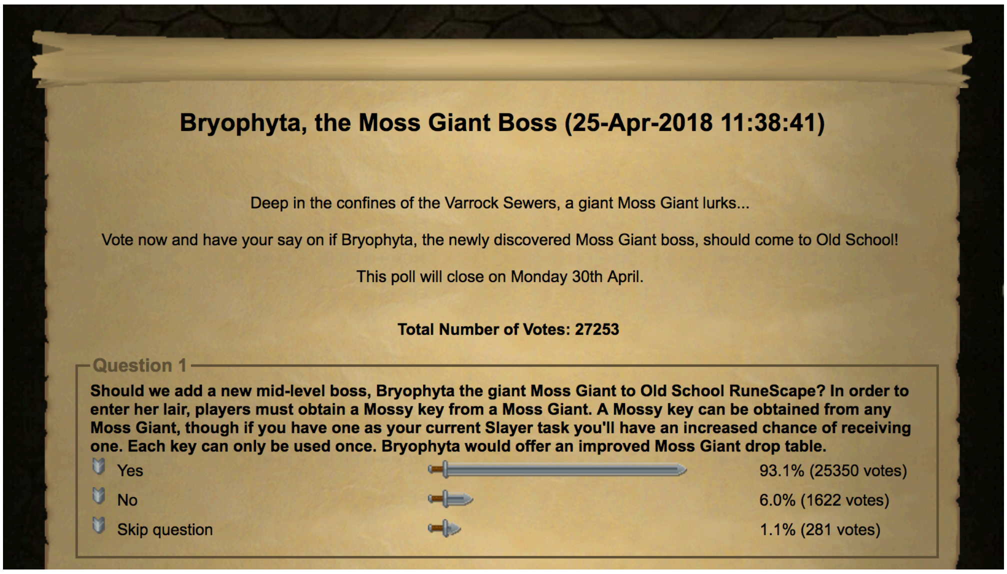
Rob Galankis, while noting problems with game development, states that it “has not embraced open source” [xxxv]. Information is valuable, and the dichotomy of the industry is such that developers rely on knowledge-sharing while wanting to maximise their own chance at success. In addition to being marred by company politics, although some find time to write blogs, most developers are not academics. Thomas Medler also warns of a prevailing fear that analytics when used in a traditional business context “sell short the potential… to create stronger connections between designers and players”.

The largest annual gathering of game developers, ‘The Game Developers Conference’ (GDC), is touted as a learning and networking event consisting of lectures by industry professionals, and is likely the most candid source of public information, with sessions streamed for free online. The quality of talks is wildly variable, and this is echoed in concerns pinned to their ‘call for submissions’, where they are “trying to strike a balance between improving the submission process for the submitter in an industry where publishing and sharing information is important but is not the main part of one's job”[xxxvi].

Despite the patchy availability of specific technical knowledge, there is an abundance of advice online in the form of retrospectives, or, for concluded projects, ‘post-mortems’. These are rather colloquial in their publication, however they all indicate towards practical applications for games analytics as a tool during development: “the game design process could benefit from analysing player behaviour to improve upon as well as validate design decisions”[xxxvii], and studies return more positive than negative repurcussions[xxxviii].

Sentiment Analysis

Sentiment analysis has become an essential tool in the games industry. It enables an automated approach to surveying the impact of product decisions through user-created content. In examples of online games, chat history and polling may be employed, however some developers rely on the publication platform (such reviews in app stores).



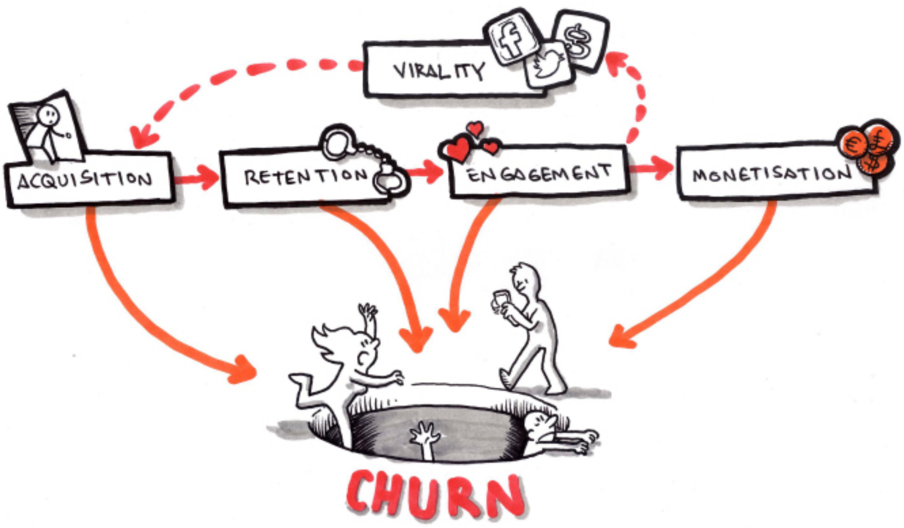
Players of the MMO RuneScape are polled before each development decision[xxxix].

Whether the data is labelled or not does not appear to matter, as general text comprehension tools have begun to match human inference[xl], and recent studies of user reviews and the intrinsic properties (‘aspects’) they reference show that “there is a strong connection between the sentiment of the aspect and the rating the reviewer provided”[xli]. The content of a review, however, is essential in understanding how specific aspects are being received. It has been observed that many users refer to mutual aspects, aiding manual investigation of reviews as a method of responding to feedback in lieu of comprehensive system, irrespective of the quantity of reviews logged.

Most sentiment analysis utilises pre-existing services, such as *x*. The most important part comes in the proceeding statistical analysis, and how these analytics are used to drive further decisions. Depending on the aspects highlighted, some companies, such as Jagex, will use these insights as a measure of whether existing content needs to be amended, and information is disseminated among producers and product owners in order to adapt release schedules. In most cases, the use for sentiment analysis is only “able to detect trends in sentiment over time as the gaming product became more mature”, meaning a static snapshot is not representative of the product as a whole.

<u>Churn Prediction</u>

‘Churn’ is the act of a player not returning to the game within an observed time period. For comparison, an ‘observation period’ is defined prior to a ‘churn period’. Churn prediction is a type of forecasting used in the games industry to predict player drop-off (and ergo supplement the quantification of ‘retention’ as a success factor).



A state machine describing the lifecycle of a player[xlii].

Predicting whether a player will churn requires feature extraction. One paper utilises feature ranking to pinpoint durations of play as timestamps as important predictors, and furthermore uses them to assert that logistic regression is the most effective machine learning algorithm to detect churn AoT[xliii].

<u>Score Prediction</u>

There have been several studies into the prediction of whether a game will be successful overall.

One by Kyle Orland uses historical data to graph estimated sales against the critical review score of a game, with the intention of determining whether strong (positive) reviews were a factor for success. He notes that when analysing metrics, there are external factors that make them inherently flawed; the source of data is biased towards specific outlets for example. He is ultimately decried by commentators for skewing results in his graph in an effort to make sense of the noise present. His results show no strong trend, and he notes that for all of his observations there are many exceptions to the rule[xliv].

Some studies attempt to build predictive models from features such as store page listings. While presenting his machine learning project 'Deep Gabe' at GDC, Alex Champanard shuns descriptive analytics presented by 'post-mortems', and emphasises using models more sophisticated than linear ones as there are "subtleties to the data these can't capture"[xlv]. He describes using a supervised 'neural network' and 'auto-encoders' to predict game scores based on previous store page listings. While this may make observations of the current market to an accurate degree, it omits that games are a living product subject to change.

All of these studies attempt to solve a problem that is arguably too complex to be captured by a single kind of metric. In the same way that a virtual physics engine can only approximate true physics through abstraction, their findings do not align with the adaptive nature of development. This is apparent as they offer few suggestions as to applicability of their findings.

Summary

It is clear that while there are many novel applications for analytics in the games industry, there are many barriers to sharing information about their use - from their grass-roots nature demanding that 'telemetry' be bespoke, to the hindrance of competition. There is an abundance of retrospection, however sources are often anecdotal. Those that are actively sharing research are for the most part developers promoting small 'passion projects' scoped in parallel to their current work, and those that are outside of the industry are at a detriment of quality information that makes their investigations either unconsidered, or too broad in scope.

This paper aims to address this by providing all parties with a common toolset to collect an abundance of data for which the links have not been publicly explored, and furthermore elaborate on these investigations by studying the more microscopic interplay between different metrics, placed into the context of a chronology of events. Throughout the investigation, care is taken to not over-reach with statements, and direction is provided in collusion with industry contacts.

3.1.3 Data Science

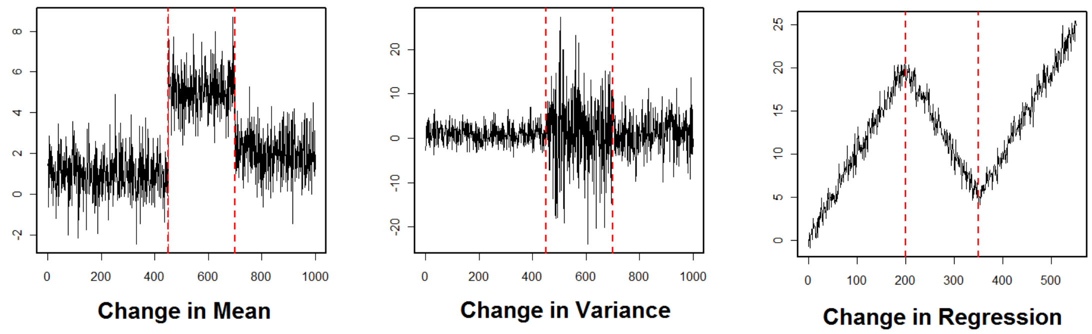
Aim

Data science is a broad term, however any description demands a level of prior mathematics and statistics knowledge. The aim here was to explore the most relevant specialised concepts necessary to conduct a meaningful analysis of the data collected, based on indicators from previous research.

Change Point Analysis

Change point analysis is the process of examining where, on a set of sequential data referred to as a 'signal', the rate of change is the greatest. It often has application on the stock market, where it is used in predictive models to forecast trends.

There are different types of signals and their reliant detectors. A *static* mean detector uses stopping criteria to indicate when the residual mean of a signal has changed. This kind of detector is sensitive to threshold values and outliers, and it therefore not suitable for something like CCU, where there is a lot of aberration.



Examples of different kinds of change-points[xlvi].

Z-scores are used in signals with seasonality to compare two 'windows' of statistics – the entire signal, and *n* most recent[xlvii]. Seasonality is described as "repeating patterns within any fixed period"[xlviii]. They make a time-series non-stationary and therefore difficult to model[xlix].

$$z_{score} = \frac{(M - \sigma)}{SE}$$

- The population has mean μ and standard deviation σ.
- The sample has mean M and a size of n. We will treat the local window as the sample.
- *SE* is the 'Standard Error'.

This reduces the accuracy of the change-point, reducing it to the proximity of the window itself (as a delay), however it is able to ignore anomalous points. In all cases, the tuning of 'hyper-parameters' is necessary.

There are many tools that can automate this process and abstract it from its underlying algorithms, such as the feature-rich 'Prophet'[I], by Facebook.

Linear Regression

Linear regression is a statistical model using in machine learning that examines the relationship between two (simple) or more (multiple) variables. In linear regression, the relationship between the dependent and independent variables is unknown, and the dependent is predicted using a 'line of best fit' that attempts to reduce the 'mean-squared error'. In a multiple linear regression model, the equation is:

$$Y'i = b0 + b1X1i + b2X2i$$

Where Y is the dependent variable, Xn are independent variables and b is the Y intercept.

There are many libraries that provide functions for linear regression, the most popular being 'SciKit'.

3.2 Industry Interview

An industry interview was conducted with Erik Lagel, senior producer currently spearheading external publishing at Jagex to see how closely his experiences matched conducted research. There were a number of talking points.

In discussing whether there is a formality to choosing new content develop[I1], Erik describes Jagex's 12-stage pipeline, or, "*filtration process*" which incorporates a wide variety of opinions from different stakeholders, including potential users. As a product passes elimination at each stage, it is "*preparing investment*".

In discussing timescales[I2], although he mentioned that this was dependent on many factors, he noted that it takes three to six months on average to produce a prototype.

In discussing the role that community plays in the process[I3], he emphatically endorsed an attitude of "*the more the better*", while lamenting the descriptions of products as 'alphas' and 'betas' – "*as soon as you make it open… then it's there*". He furthermore speaks of success occurring "*as long as the vision of the game is capturing the necessary audience aspirations*". This is a service mentality that places the entire purpose of a product on what he calls "*player obsess*", and is an exaggerated reflection on prior study. Ultimately, he praises 'early access', as "*the earlier you can actually validate that your idea… has got a resonance with your target audience, the better you are because you can build on this*".

Finally, in discussing whether the process of validation is data driven[I7], he preferred the term 'data supported', arguing that this protects the creative process. He cautioned against the difficulty of measuring intent in what he describes as an industry with "*extreme volatility*", advising instead that "*the best way is to actually feed on the evolution of others*".

In summary, the discussion with Erik was advantageous as it provided several leads for explicit metrics to study, as well as validating in greater detail the the prescient points unearthed during research.

3.3 Existing Services

Analytics Services

There is a plethora of third-party analytics services that exist to service game developers' decisions throughout the NPD lifecycle. They offer a number of competitive feature-sets, with most attempting to achieve greater depth than the simple event tracking of free alternatives.

'Game Refinery' focuses on qualifying game features, and at this level of granularity charts user progression through a game in addition to the performance of similar features across other products[li]. 'DevToDev' lacks this feature but places greater emphasis on data visualisation and financial reports[lii].

The services appear to only be limited only by their integration costs, however finding a truly unbiased account of their efficacy is not possible. Nonetheless, their existence illustrates the desire for games companies to understand the impact of their decisions during development.

Platform

'Steam'[liii] is a digital distribution platform developed by 'Valve Corporation', and is the largest of its kind in the PC gaming market, particularly for 'early access' titles (which now exist on other stores such as Xbox). In exchange for a fee and percentage of sales, it provides industry professionals with publication, marketing, and further full-cycle analytical and user engagement tools.

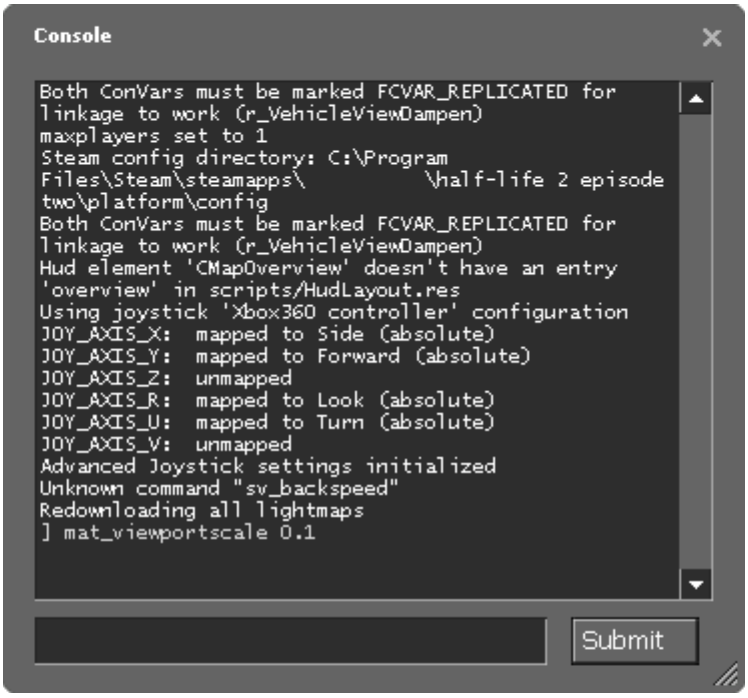
Access is limited, as many of these features are restricted to accounts that own active products. Furthermore, the interface itself is constantly changing[liv]. Speaking with senior product designer Alden Kroll of the ongoing work:

"*We will conduct our own business-information as needed to figure out whether something is working or not, but also incorporate a lot of user feedback from players and developers. We're not really driven by business cases, but instead driven by what are players asking for, and what we can do to make them happy. If the review system isn't helping players make informed purchase decisions, then it's something we want to change. Even if it 'feels' wrong, then that could be enough for us to spend time on making it feel more right and fair.*"

Resultantly, while the the information itself is immutable, all sources of information are potentially volatile. Despite these caveats, the ecosystem provides enough potential for exploration to justify it as the primary vessel to satisfy this paper's aims.

APIs

Steam is limited insomuch as their public facing systems only provide snapshots of the current internal state of their systems. Due to this limitation, several third party developers have open-sourced systems that track this data over time, and their APIs have been the primary sources for data collection via scraping. They do however provide historical review data. This is intended to be used by their front-end pages, however it is publicly accessible and documented[lv]. The



The Steam console accessible in the client's debug mode provides an observable stream of events.

SteamDB[lvi], tracks all metrics available through Steam's console since 2011. It is the most comprehensive source of information for all available metrics, however it is limited insomuch as it employs DNS protection to prevent scraping. Since they open-source their tracking software, it is assumed that direct API or database access is denied to the general public to prevent competition. This can only be legally circumvented with permission, which was attained for this paper.

SteamSpy[lvii], introduced 2012, tracks aggregations of information from a mixture of Steam and SteamDB. It is beneficial insomuch as the data is combined ways intended to demonstrate connections between metrics, providing at-a-glance insight. However, it is limited insomuch as without an account with elevated privileges, the data range on all graphs is capped. Primarily, it provides an index of games by search 'tag', therefore providing a complete list of products under the 'early access' release model. Second, it provides a proxy to a preformatted account of each product's store listing, which is more convenient than Steam's own representation.

Privacy Policy

The use of data in this project complies with the terms and conditions of the platforms from which it has been derived[lviii], and furthermore legislation on data protection [lix]. There is no personally identifying information accessed, and therefore no ethical issues have been discerned.

4 Data Mining

4.1 Technologies

The following series of technologies have been used in tandem to achieve this project's goals.

Python

Python was chosen as the *project's base language* due to its popularity[lx] in the field and subsequent abundance of web-scraping and scientific libraries (and their accompanying educational resources)[3.2.1.3]. By enforcing a standardization between the data-mining architecture and data analysis, library functions and models were shared to reduce boilerplate code.

JSON

JavaScript Object Notation (JSON) was chosen as the *data interchange format* as it most closely aligns with the return format of the various APIs in use, in addition to the preferred storage format of the document-oriented database. While this is less common than CSV, it resulted in easier transformation during collection and manipulation.

Microservice Architecture

I chose to build a 'microservice architecture' in order to perform the *data-mining*. It is a service-oriented architecture that comprises of a collection of loosely coupled services that interface via APIs and messaging layers, with each service performing a single role.

This style is advantageous as it promotes modularity and ergo extensibility in the codebase and development. The services are independent, granular, and can be parallelised. This is preferable to a series of otherwise monolithic scripts, for which performing complex tasks is usually only attainable at the expense of clarity. Therefore, the entire problem could be broken down by the tenet of divide-and-conquer, and in a properly designed system, at any stage any technological choice or service can be replaced without consequence, removing hard dependencies.

There is a notable disadvantage to this design that requires mitigation: elements of threading/synchronicity in this distributed system can result in a lack of fault tolerance without monitoring and orchestration, which was addressed via additional services.

'Flask' was used to host services acting as web applications (those with an API or template-rendering) due to its compatibility with the automatic documentation tool Swagger, which provides an intuitive interface to the system concurrent with my objectives.

‘RabbitMQ’ was used for messaging to transfer directives between microservices as it is a performant, general purpose protocol.

Document-Oriented Database

‘Couchbase’ was chosen as the database for this project for a number of reasons. Primarily, as a non-relational storage solution it can handle missing or variably sized information, making it suitably flexible for the type of data collected.

Furthermore, it has undergone a series of updates since its previous incarnation as ‘CouchDB’ that have provided innovative ways of accessing and manipulating data. Through ‘N1QL’, an ‘SQL-like’ query language, data can be indexed, altered and retrieved. This can be accessed via a web interface, which provides a ‘workbench’ of tools (such as query history and performance monitoring), or via a web API which is also accessible using a Python software-development-kit.

Containerisation

The entire project has been ‘containerised’ for distribution using ‘Docker’, a virtualization software that allows for deployment irrespective of the underlying hardware. Theoretically, the project can be run on any machine with this software installed.

This has been done to enable ‘cloud’ deployment to a remote machine (and ergo universal access to the data), and promote reproducibility for interested parties wishing to copy the project’s conditions with a helpfully small set-up cost.

Analysis

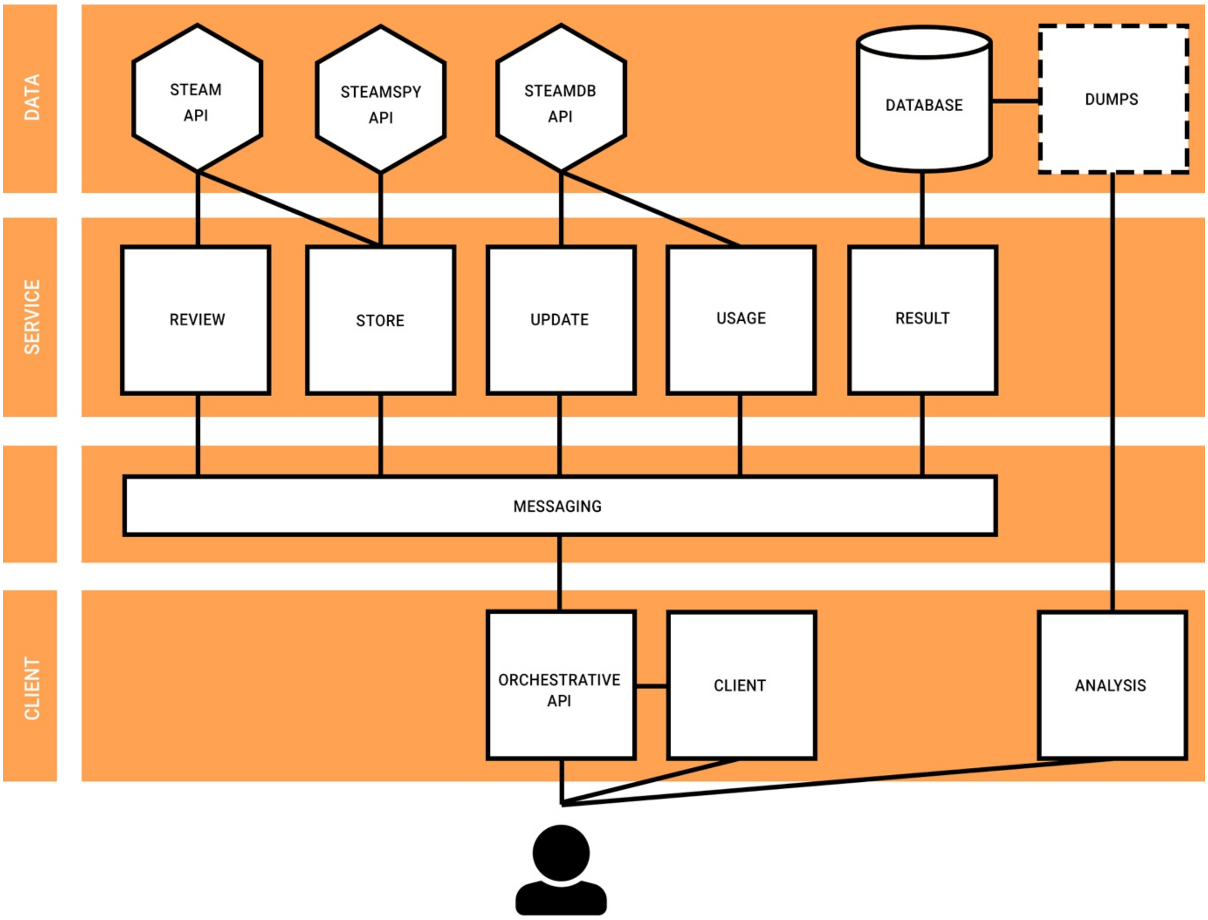
‘Jupyter’ is a web application that provides an interactive interface to a combination of ‘markdown’ and Python described as ‘literate programming’. It has been used to document the statistical analysis of the paper in a verbose fashion, charting the ETL with accompanying narrative text. This is advantageous as it allows for the bulk of the project’s work to be centralised and versioned, with the prose itself expressive and dynamic[lxi].

Inline graph plotting has been achieved using ‘Plotly’, aided by several functions from the ‘Numpy’ and ‘Pandas’ libraries, as they depend on models provided by each other.

4.2 System Architecture

This section describes how the architecture was designed and built. The individual components work together to fulfil FO2.

High-Level Overview



System architecture diagram showing major components and semantic layers.

At the *data layer*, the APIs the system scrapes are shown, along with the database the processed data is persisted to. *Dumps* is not a running service, but rather a static snapshot of subsets of manipulated data to be used to remove the dependency of *analysis* on the live database.

At the service layer, the *review*, *store*, *update* and *usage* microservices monitor their respective work queues awaiting instruction. Upon receiving a payload containing a valid product ID to scrape, they make requests to their respective APIs, storing the results back on *result*'s work queue. *Result* is responsible for storing the state of jobs (including errors that may occur), and processed information in the database.

The *messaging layer* is a single pre-provided RabbitMQ service. Each service could also have their own compartmentalised messaging service, however this is more useful in more complex systems.

The *client layer* provides a public interface to the system as a whole. The *orchestrative API* hosts its own documentation, and is responsible for queuing a job as a compound task. The *client* service provides reporting functionality to monitor the internal state of the system. The *analysis* service hosts the Jupyter kernel used to create notebooks. The delivery mechanism here is the browser, and for the orchestrative API, any web-enabled technology.

Each layer also supports direct access via administrative interfaces.

Database

Since the database does not have a relational structure, it defines the concepts of ‘buckets’ – separate free-form data stores.

name ▾	items	resident	ops/sec
job	977	100%	0
review	1,533,...	56%	0
store	977	100%	0
update	33,934	100%	0
usage	977	100%	0

Bucket list in ‘Couchbase’ interface.

There are buckets for all discrete data types, in addition to an overall job state which tracks for a product *n* service states.

```json
{
  "product_id": "108600",
  "review_finished": true,
  "store_finished": true,
  "update_finished": true,
  "usage_finished": true
}
```


An example job state document.

Messaging

Each microservice on the service layer has its own message queue. The *orchestrative API* accesses each. As they all rely on the messaging layer, although some alignment has to be enforced in the configuration file for each to ensure proper functioning, they are independent from one another.

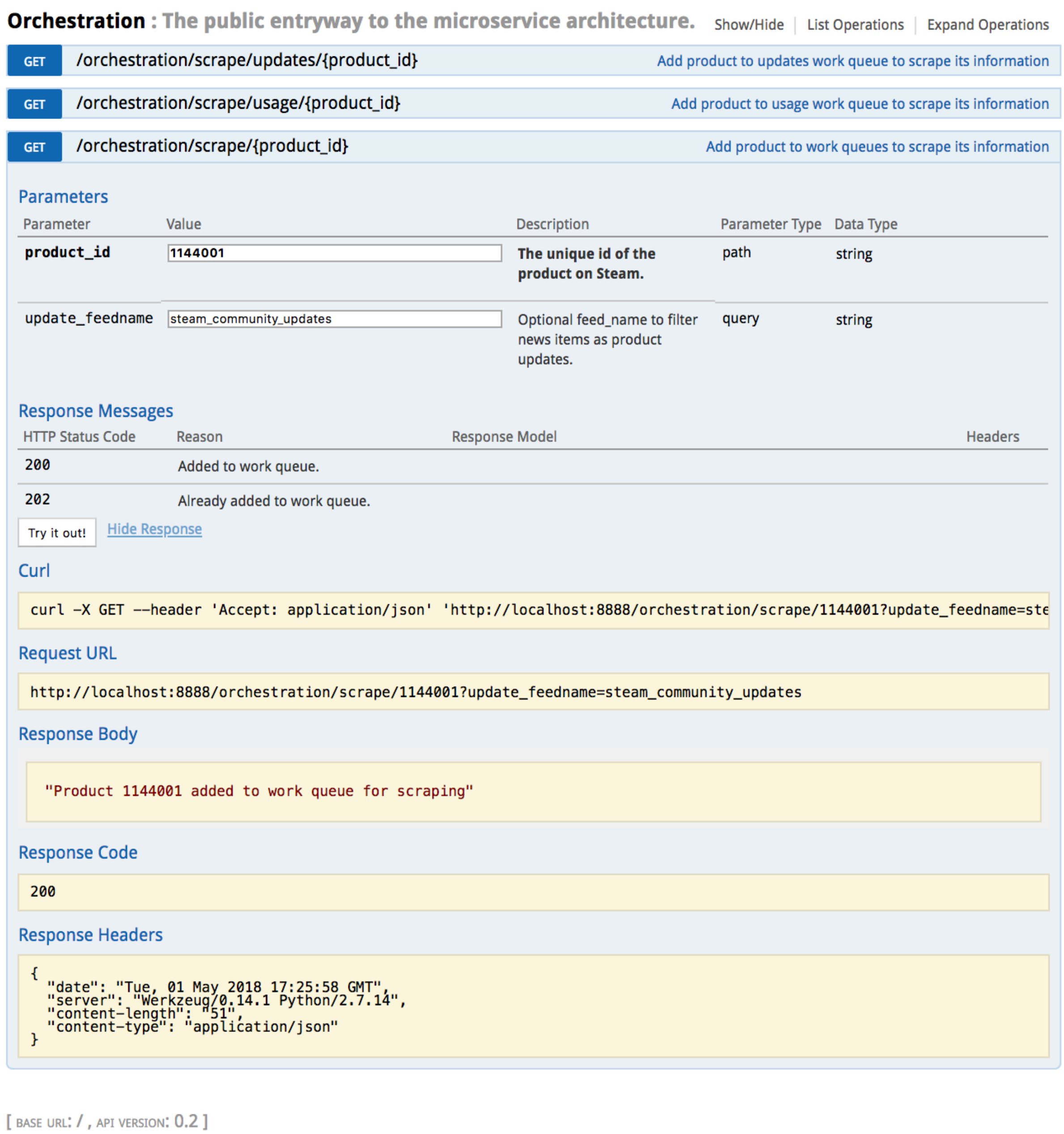
Overview			Messages			Message rates		
Name	Features	**State**	**Ready**	**Unacked**	**Total**	**incoming**	**deliver / get**	**ack**
global_error	D	idle	0	0	0			
work_review_queue	D	idle	0	0	0			
work_store_queue	D	idle	0	0	0			
work_update_queue	D	idle	0	0	0			
work_usage_queue	D	idle	0	0	0			

Each work queue as shown in the 'RabbitMQ' interface.

The queues themselves operate as 'round-robin', meaning that services that have been horizontally scaled during deployment are essentially load-balanced.

API

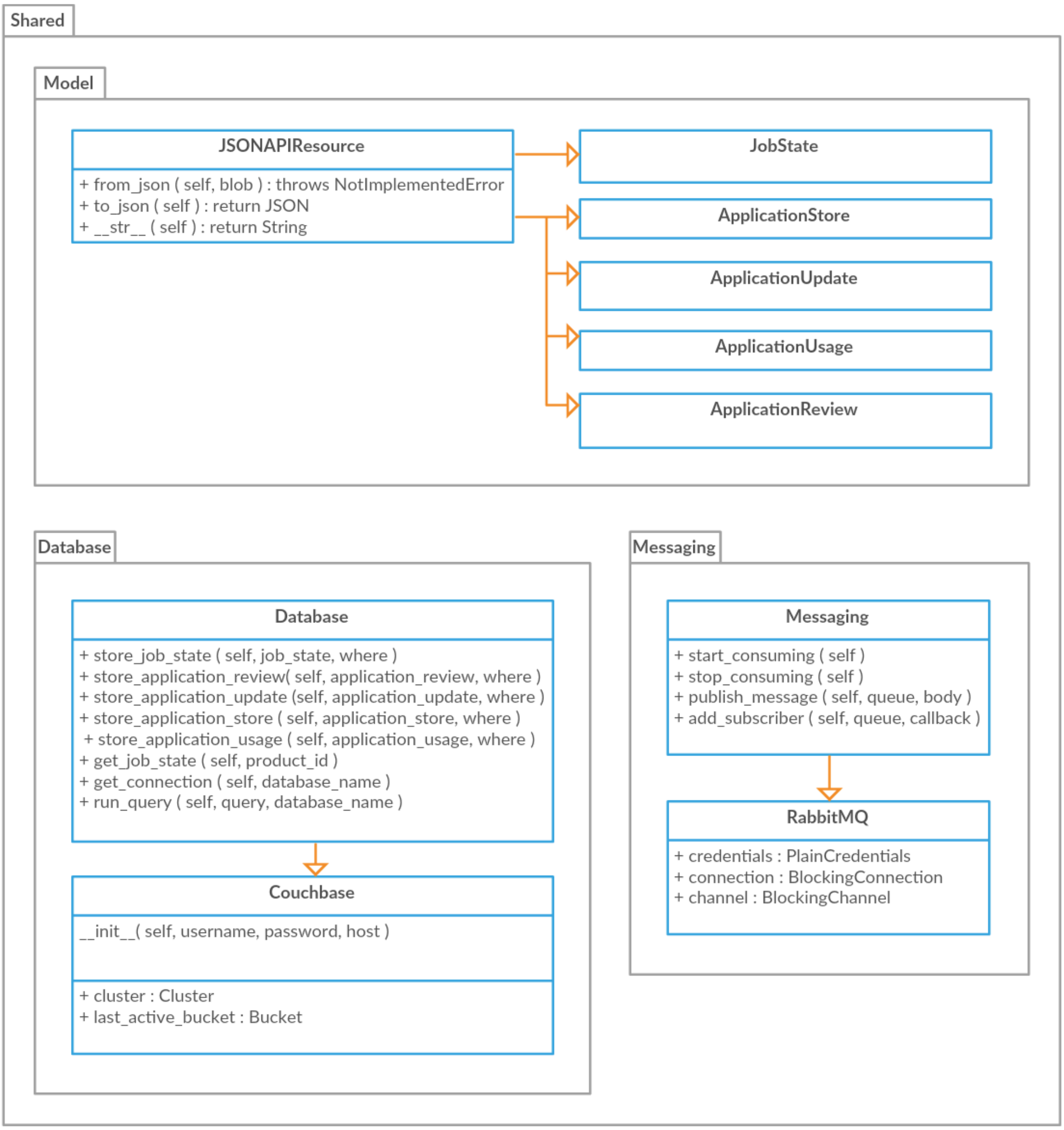
There are a number of endpoints exposed by the *orchestrative API* in order to queue jobs. The most all-encompassing is: `/orchestration/scrape/{product_id}`, which requests that all services begin scraping. Each individual service can also be prompted, for debugging purposes. The return value indicates the success of the response.



'Swagger' documentation describing available API endpoints.

Shared Library

Data returned from different public APIs may be inconsistent and subject to change, as may the different underlying technologies used for storage and message-passing. To allay this, a shared library has been created to define interfaces with a polymorphic structure that describe each unit of data and common operation. Each service imports this where necessary to convert between raw and memory-managed representations to perform operations upon.



Class diagram for the shared package and its respective files.

Deployment

The system can be deployed locally or to the cloud depending on the pre-configured 'docker machine'. Each service defines its own 'Dockerfile' configuration file which describes how it should start-up and shut-down, in addition to security profiles and technology-specific settings. Further configuration in the way of dependencies, networks (port forwarding) and volume storage is defined in the over-arching configuration file 'docker-compose.yml'.

4.3 Collection

This process in effect tested the efficacy of the system. A list of all ‘early access’ product IDs provided by SteamSpy was iterated over in a worker script, calling the ‘scrape’ endpoint for each with a delay in between. In doing so, API restraints were circumstantially avoided; timeouts were employed to avoid rate-limiting, and browser-spoofing using the ‘cloudflare-scrape’[lxii] library was necessary to bypass DNS protection (with permission).

While by no means part of intended experimentation, the system’s hyperparameters needed to be adjusted in order to ensure that the entire process of collection was reasonably quick. Initially, without scaling any services, the ‘review’ service was still computing after 8 hours, which was deemed inefficient. Furthermore, the allocation of RAM per bucket needed to be adjusted proportionate to its number of operations per second over time; therefore, ‘review’ and ‘update’, which entailed dealing with pagination, were favoured over ‘store’ and ‘usage’, which only required one web request each.

The individual services themselves do not consume much resource on the host system (~60mb while idling with one of each deployed), however since the length of information to be retrieved by some services is variable, bottlenecks are tough to predict ahead of time. In order to determine how many services to deploy, those with the highest load were scaled upwards (while running) incrementally until their message queues maintained an equilibrium in depletion as jobs were queued. Theoretically, the fastest completion time would be achieved in a system with *n* services for *n* products, however this is unrealistic.

After scaling to 20 ‘review’ services, 10 ‘update’ services, and two ‘store’ and ‘usage’ services, information for all products took 4 hours, 32 minutes on a machine with 4GB RAM and 2 virtual CPUs. This operation was repeated on multiple machines throughout the duration of the project, and multiple backups of raw information (679.96MB raw, 1GB with indexes) were taken.

4.4 Pre-processing

Combining Sources

Some units of data drew from multiple sources due to omissions and discrepancies in each. For example, to retrieve the kind of information displayed for a product in its online store interface, the Steam API does not provide user score; this is accessible through SteamSpy. Furthermore, the release date of a product provided by Steam is the intended date of release by developers. This may not align with the product going 'live', and is more of a denotation for marketing purposes. Therefore, the earliest point at which members of the public can purchase and engage with it is derived from the date on which SteamDB began reporting daily users for it. This amalgamation of objects provides a truer representation of each product, with the caveat that each field requires explanation to dispel ambiguity, in addition to synchronicity among sources. While the absolute correctness of the data cannot be guaranteed, extensive manual testing was undertaken to verify that it was correct enough to suit the project's goals, and the system is flexible enough to incorporate multiple different representations of the same categorical feature within a single object.

Culling

After collecting as much data as possible for the products in question, those with incomplete information were removed. This included those that were unreleased (7), had no reviews, updates, or usage (313), and furthermore those with fewer than 20 reviews or two updates as a conservative threshold for enabling the examination of iteration over time (486). This reduced the number of products in the system from 1,783 to 977.

Indexing

To improve query speeds over the larger buckets, especially during join and sub-selection clauses, global primary and secondary indexes were constructed on document fields. While this sacrifices some disk space, this reduces the order of operations, accruing worthwhile performance benefits.

To exemplify this, we take the query:

```
SELECT COUNT(*) FROM `review` where `voted_up` = TRUE
```

With just a primary index on the document's ID, this takes 11 minutes to perform due to having to iterate over the entire set, and may result in unexpected behaviour.

With a secondary index on the 'voted_up' field this takes 2 minutes to perform.

With a conditional secondary index on the 'voted_up' field having the value of 'true', this takes just 13 seconds to perform.

Munging

Data ‘munging’ was performed to extrapolate aspects of the data into new fields, or create new groupings to serve the analysis more efficiently. This took the form of a number of scripts with a mutual structure:

```
product_ids = db.run_query(‘select product_id from job’)

overall_results = []

for product_id in product_ids:
        # Construct a new representation
        new_object = {}

        # Run a number of queries on various buckets
        new_object.new_field = db.run_query(‘’)

        # Add it to the overall result
        overall_results.append(new_object)

save(overall_results)
```

Pseudo-code for worker script.

In doing this, missing information was also able to be determined and filled. Some fields listed as ‘None’ or ‘NaN’ were converted to 0, and missing days when constructing timeseries were replaced by days with zeroed metrics.

5 Analysis

5.1 Hypotheses

There were a number of initial hypotheses building towards the overall aim that helped divide and provide further stimulus for the investigation. These were split between examining whether a greater number of product updates incur better review scores[RX, UX] and CCU[CX]. Each step represents an attempt to study if and in what circumstances developer engagement impacts success metrics of user engagement over time, and can be quantified.

5.2 Constraints

The system is constrained to a set of historical interactions in order to focus investigation. Abstractly, they capture cyclical developer and user feedback (engagement) throughout the ‘early access’ lifecycle.

A ‘user’ represents somebody who has purchased the product, and encompasses a set of interactions within this system including:

- Using the product
- Publishing reviews about the product

A ‘developer’ represents somebody with control over a product’s lifecycle, and encompasses a set of interactions within this system including:

- Publishing the initial product listing
- Publishing updates to the product itself

5.3 Summary

Model Definition

Product data was collected from four discrete sources, which represent different categorical features sufficient to describe the relevant interactions.

- Product listing
- Product reviews
- Product updates
- Product usage

More columns were collected for each than were actually used during analysis. Some were used heuristically, others became useful over time, and some that were missing were added later. By collecting multiplicitous information, the potential use cases for the same data set have been increased.

Data Types

Timestamps are for the most part Unix epochs, tracking time elapsed in seconds since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970. This is useful because as whole numbers they can be easily compared and converted. The smallest units of time in steps are days (86,400).

Listing

A 'listing' discerns it from other products. They are stored in the 'store' bucket (the discrepancy in nomenclature is explained in the source code). The fields are considered immutable insomuch as there is no historical account of modifications available.

```json
{
  "product_id": "284850",
  "date_released": "8 Nov, 2013",
  "genres": [
    "Indie",
    "RPG",
    "Simulation",
    "Early Access"
  ],
  "is_free": false,
  "metacritic_score": "68",
  "name": "Example Product",
  "owners": 872046,
  "players_average_forever": 1231,
  "players_forever": 837417,
  "players_median_forever": 391,
  "price": "1499",
  "score_rank": 65,
  "user_score": 86
}
```

Each may have multiple ‘genres’, (also ‘categories’ or ‘tags’).

Review

A ‘review’ represents a pre-classified unit of sentiment, one per users. It also contains some flattened information about the user’s profile (usage habits). A review can only be submitted once by a user who owns the product.

```json
{
  "product_id": "284850",
  "author_id": "76561682049575891",
  "author_last_played": 1475862888,
  "author_num_games_owned": 90,
  "author_num_reviews": 3,
  "author_total_playtime": 623,
  "date_created": 1399060157,
  "language": "english",
  "received_for_free": false,
  "recommendation_id": "16000027",
  "review_length": 122,
  "voted_up": true,
  "votes_funny": 0,
  "votes_up": 0,
  "written_during_early_access": true
}
```

Update

An 'update' is a publication to the community news feed by a developer. These are usually but not exclusively used to promote awareness of the availability of new content. They may also represent external factors not always captured by the system, such as the participation in promotional events.


```
{
  "product_id": "284850",
  "date_created": 1387226838,
  "feed_name": "steam_community_announcements",
  "update_id": "1999102271521259752"
}
```


Usage

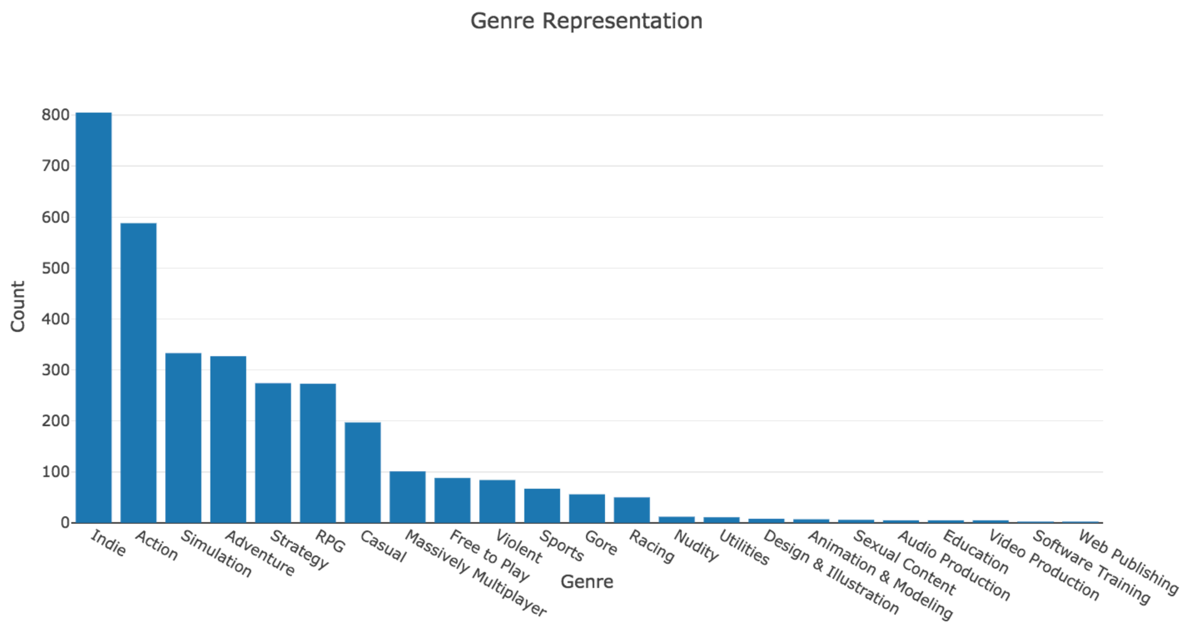
'Usage' refers to active time users have spent using the product. This is represented as peak daily CCU since the product's release. It has been flattened to promote readability, thus individual days must be indexed using a function of the start time and step.


```
{
  "product_id": "284850",
  "start": "1366834400",
  "step": "86400",
  "values": "[1, 16, 31, 46, 61, 76, 91, 106 ... 13]"
}
```


Overview

The dataset contains a total of 977 product listings, 977 corresponding usage documents, and an accompanying 1,533,188 reviews and 33,934 updates.

In terms of *representation*, there are 23 genres. A product can have up to a maximum of 5 listed genres; these are loose descriptors. There are 942 video games and 35 software products. Products that have since migrated from 'early access' are not included.



Release dates of products in the dataset spans from the one released longest ago, on 08/11/2012, to the most recent on 09/02/2018 (5 years, 3 month and 1 day). Data for all products extends to 27/02/2018 thanks to usage information – this was the date of collection.

Review counts for products range from 20 (constrained) to 139,876. The average is 1569, and the median is 122.

User scores (*sentiment*) for a product ranges from 0 to 100 as a percentage ratio of positive to negative reviews. The average user score is 72, and the median is 75.

Update counts for products range from 2 (constrained) to 490. The average is 35, and the median is 20.

Although *price* is not a focal point, it is worth noting that the dataset contains 100 free products (not due to sales).

5.4 Known Factors Influencing Data

Since a closed system has been constructed it is important to consider other important influences on the data that may influence findings; this is not necessarily exhaustive. It is likely that in some cases, insights are a result of multiple factors of variance that prevent correct feature compression, and therefore statements made about the data are prepended by the notion that their apparent cause is not mutually exclusive.

Price

One of the contributing factors to the success of the 'Steam' platform is its notoriety for consumer-friendly promotional events. Resultantly, the prices of product listings vary significantly over time, even during early access. While products in early access rise in price at intervals relative to their convergence towards a completed state, their participation in sales at the behest of marketing is evident in many products that see temporary reductions in their base prices immediately upon release. The 'Steam' store interface categorises these deals using tags, and at the time of publication lists 83 early access products on sale, ranging from 10% to 90% discounts.

These sales undoubtedly influence the metrics within my system, as they result in an increase in ownership that enables higher review counts and CCU[lxiii].

A game on sale benefits enjoys exposure on the store-front.

For the scope of this paper is was decided that price would be too complicated of a metric to consider for the value it would incur, for a number of reasons. The financial targets of each publisher is unknown, and therefore it cannot be used as a factor determining success to suit the primary aim. Additionally, the locality of other metrics is global (and presently unavailable by territory), and therefore a greater effort would have to be made to amalgamate the price variations for each currency, of which there are 40; in doing so, links would be much more tenuous to prove.

The impact of this decision has been mitigated by considering the probability that changes in the timeseries analysis were the result of an update.

Seasonal Effect

The video game industry exhibits a large degree of seasonality in product engagement, at both a weekly and monthly level.

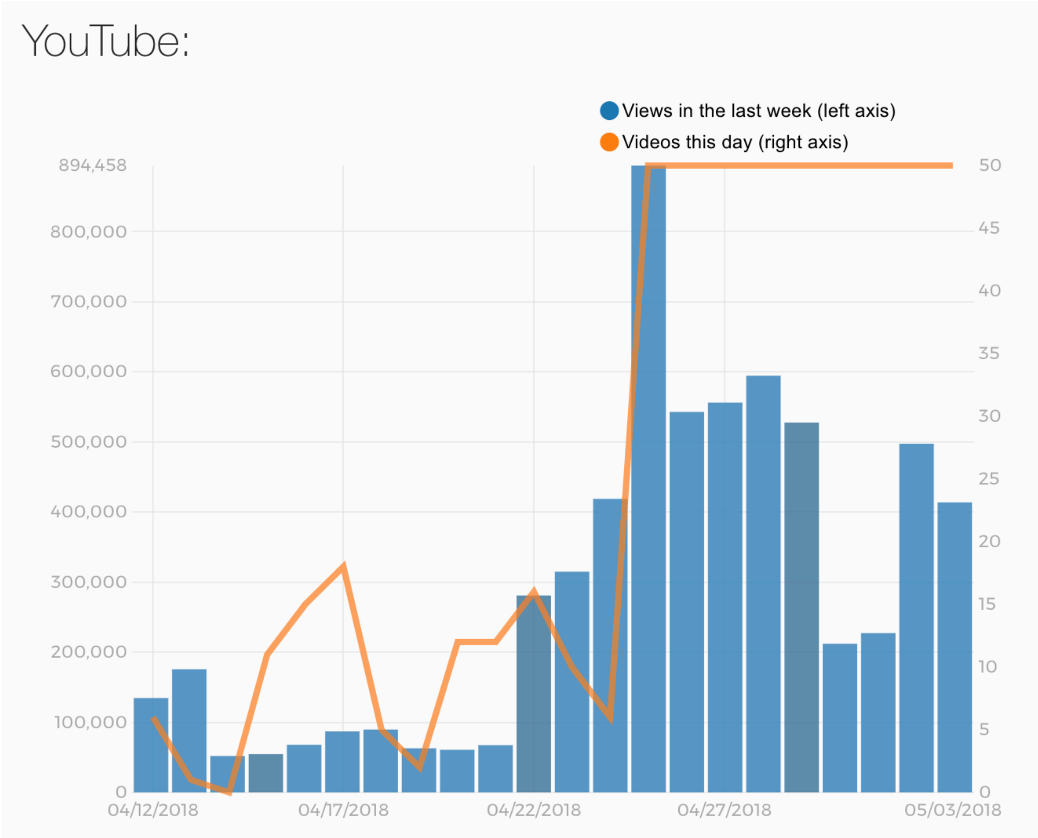
CCU lapses when the primary demographic for a product are asleep or at work. Furthermore, more engagement is seen during public holidays, particularly during December[lxiv].

Care has been taken during analysis to consider dates at which interactions have occurred – every metric is timestamped, and therefore the context for its submission has been considered and aberrations have been noted. In most instances of specificity, the active development time of the product in question spanned more than a few months, and the average age of products in the system is 1.6 years, which was sufficient enough to attest to hypotheses for times when seasonal effects are less prevalent.

Virality & Marketing

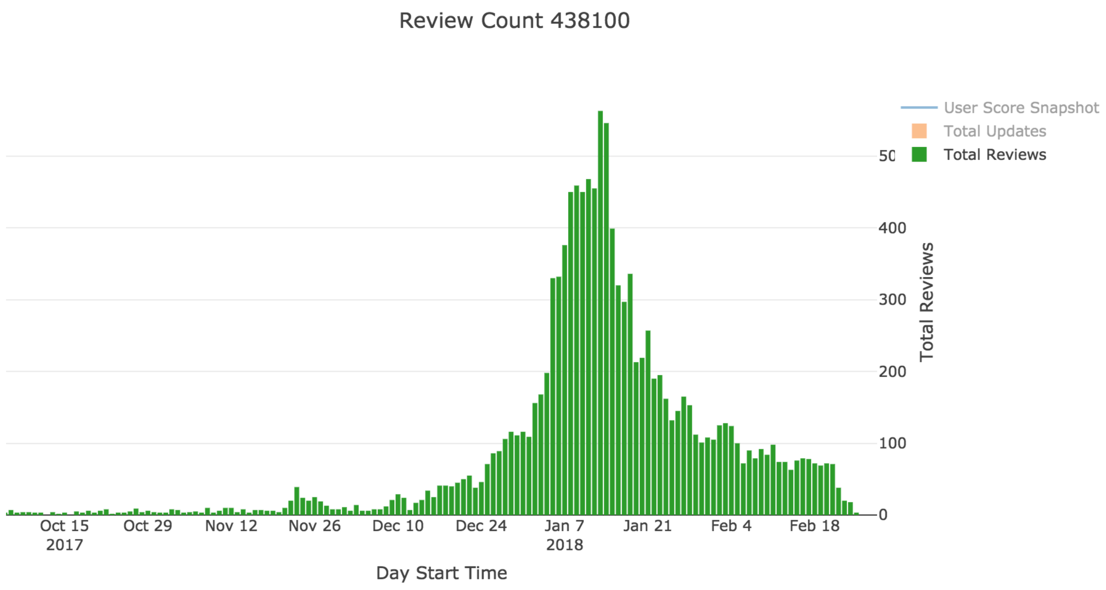
Uncontrollably and unpredictably, a combination of the above and more may help lead to a product going 'viral' and attracting more attention than another. In addition to traditional advertising, the games industry largely benefits from 'community influencers' driving traffic by spreading demonstrative footage.

This is an active area of study, and the API sources used in this project have recently begun tracking the dates of popular YouTube videos featuring products against their metrics. It is worth noting that this effect is not necessarily positive – due to this transitive association, products with attachments to controversy may be unfairly subject to collaborative efforts of defamation through a process dubbed 'review bombing', a form of online 'brigading' where users flood a product with negative reviews irrespective of its content. This too is under active investigation, however efforts by the 'Steam' BI team employing anomaly detection to prevent this are in their infancy.



Online exposure is now tracked for products on SteamSpy.

It is evident which products in the data set have attained virality - the top 10 most popular products collected represent 45% of all reviews, and timeseries graphs show a 'snowball' effect towards an elevated state of interest. These products were useful due to their wealth of information; they make the dataset itself 'big'. They are in affect, the successful cases to be replicated, and this distinction has guided some aspects of the investigation.



A product that achieved virality over the Christmas period.

However, since the reasoning and impact behind marketing success is partly composed of elements not tracked by the system, efforts have been made to either omit or reduce the influence of outliers at both ends of the spectrum of popularity during investigation (usually using logarithmic scales).

5.5 Examining Metrics

Context & Purpose

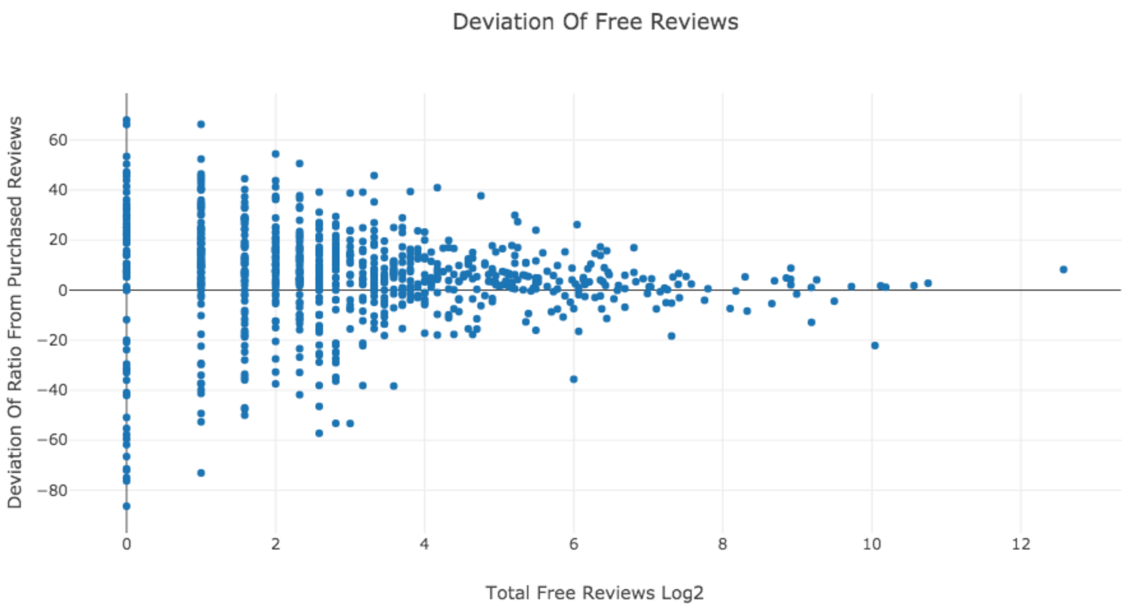
Since metrics were accessed programmatically, the sequential list of each stored in the database does not necessarily match that which is represented in the web interface. Some are therefore less plainly factual than others. For example, the metric of user score is derived from product reviews, whereby users are required to make a binary decision as to their recommendation. This is useful, as it presupposes rigorous textual sentiment analysis, however when attempting to utilise this information we must still consider the context for its original submission – reviews are able to be viewed by other users in a myriad of ways, and therefore may not be equal in influence. Therefore, it was deemed worthwhile to posit patterns in input which may affect the integrity of atomic elements of data, before relying on them in further, more complex investigation. Additionally, trends found could be exploited by developers to target specific users during the lifecycle. The web interface itself was used to identify the most notable features of reviews.

An example of a product review.

Here, we see the *author's total playtime*, whether the product was *received for free*, and the number of other *users who found it helpful* have all been emphasised. This mixture of habitual and social measurements raised a number of questions. It is tough to discern whether they are a typical consumer or a professional critic, so the the notion of personal motivation can be eschewed by looking at these statistics.

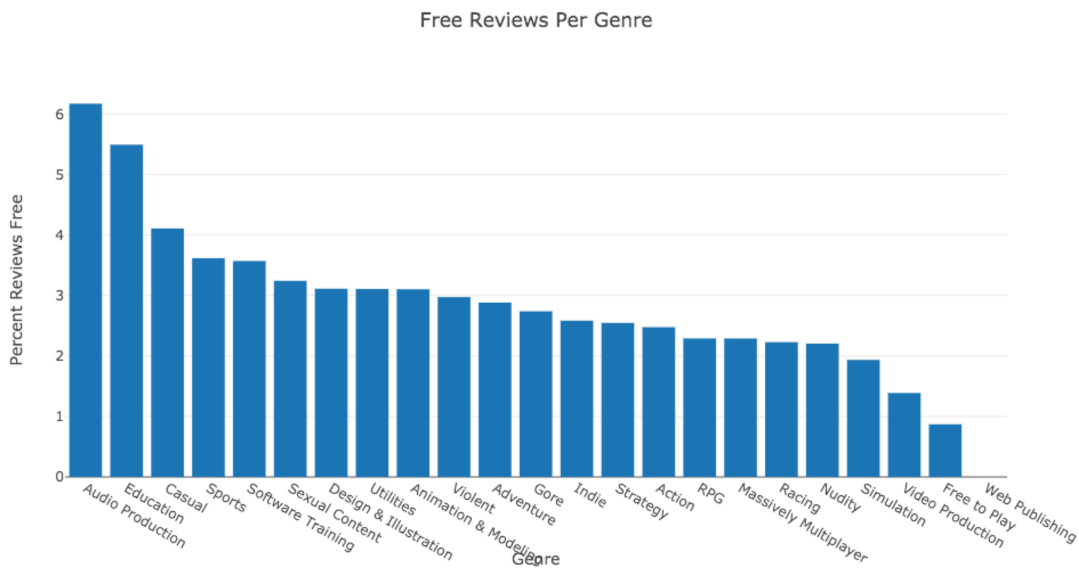
Gifting

Firstly, a positive bias was discerned for users who received the product for free[R1]. This was immediately apparent looking at simple totals. Of all reviews, ~97.58% are purchased. Of purchased reviews, ~68% are positive in comparison to free reviews, where ~78% are positive. This appeared to suggest that free reviews are biased towards positivity. This was verified by looking at the deviation in ratios between purchased and free reviews for each product. An intermediate index was constructed for every product with a price greater than 0 and at least one free review. Analysing this subset of 801 products showed that on average, 8% of all reviews for a product are left by users who received it for free. Pointedly, the average deviation in percentage of positive free reviews against positive purchased reviews was 5.18%. Since the deviation may have been drastic if the number of free reviews was too small, this was graphed to gain a clearer understanding.



As expected, products with less free reviews have wilder deviation from the purchased percentage. However, the number of products for which this deviation is positive (570) is far greater than the number for which it is not (217). Therefore, although the intensity of the trend is minor, it is reasonable to state that free reviews are more positive. The trend may be dampened by the awareness that the context of their purchase will be highlighted in the UI.

Finally, it was examined in which circumstances this is most applicable by studying for which products free reviews are left the most: software and casual games.

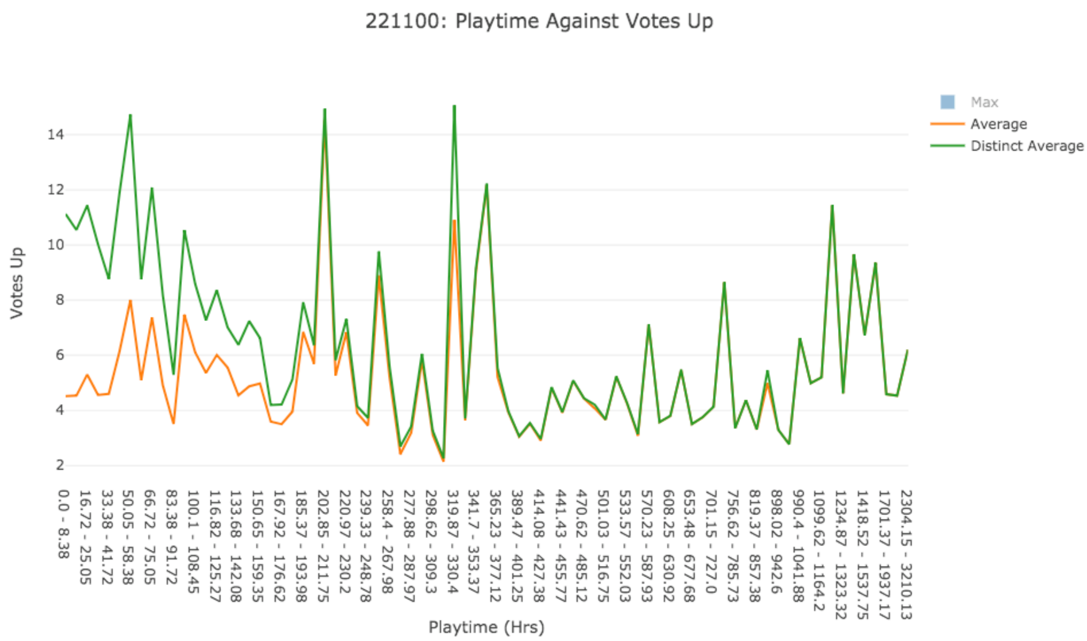


Playtime

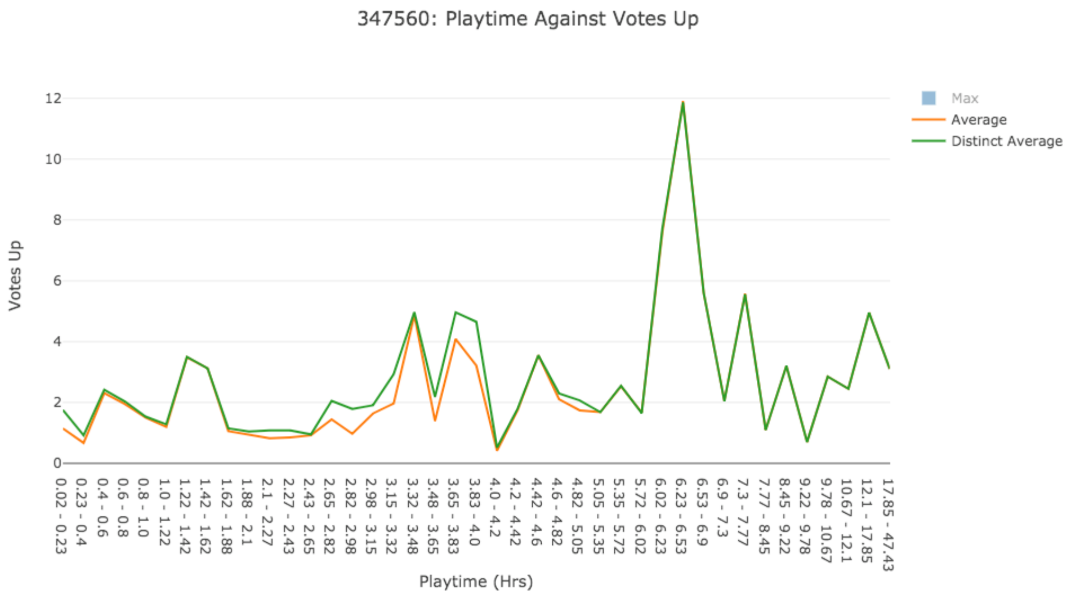
No positive linear correlation was found between the length of time spent using a product and its 'helpfulness' as flagged by the community[R2]. Since these reviews are bubbled to the forefront of the UI by default, and user score is derived from all reviews, a bias here would have indicated that this metric is flawed as the recommendations of these select few are more influential in informing user purchasing decisions and subsequent developer reaction. The initial expectation was that a user with an amount of logged hours at the higher end of the spectrum would receive more upvotes, however this relies on the poster's ability to leverage their experience into creating a more meaningful review.

A twofold barrier to this investigatory step was that the vast majority of reviews for popular products go unseen, and those that are seen but are deemed unhelpful are not able to be tagged as such. Therefore, a distinct average was taken that considered the number of upvotes in only once in each evenly distributed time band across a histogram.

For the most popular product in the system, an open ended MMO, it appeared that there were peaks in playtime for which subsequent reviews were useful, however exorbitantly longer playtimes were actually much less useful. Helpful reviews occurred in almost all time bands.



The notion of ‘completion’ was then considered as a counterpart. For a linear story game, its completion time of 6 hours (derived from community consensus) correlates with peak helpfulness. The noise here again provides evidence suggesting that there is no merit to the original suggestion, however it did give onus to exploring the link between completion and helpfulness for the entire set[R3].



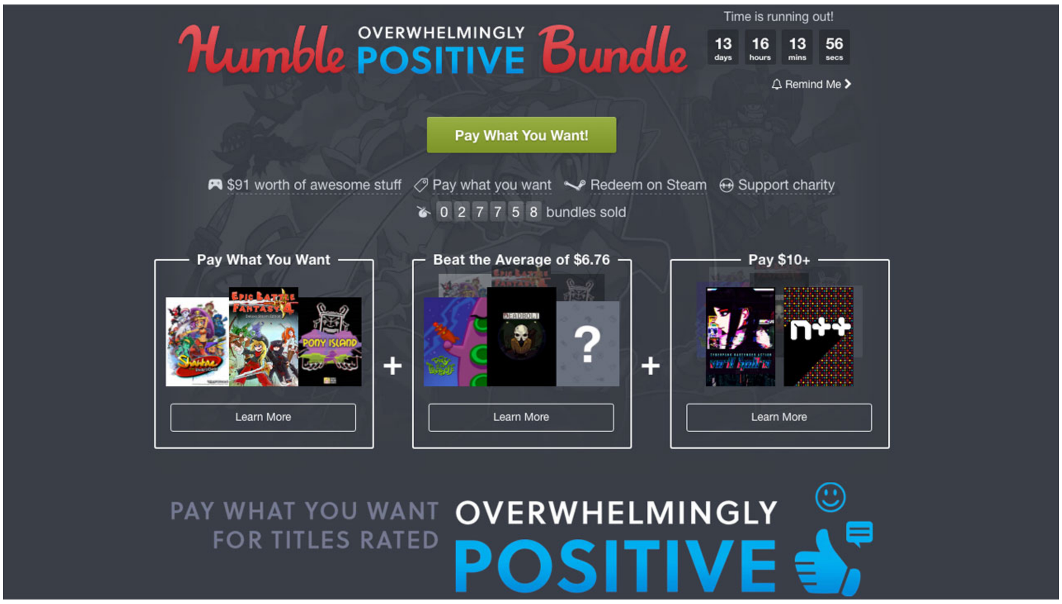
The peak review time was calculated for every product, with the average being 50.15 hours, and the median 5.18 hours. The median roughly aligns with the completion time for most games on average[lxv]. ‘Completion’ is a concept with many different definitions[lxvi], and it is therefore hard to draw a concrete connection, however this does appear to suggest that the best reviews are left by users who have finished using the product.

Helpfulness

Lastly, it was determined that while the most helpful reviews exhibit interesting patterns, they display no strong observable bias either way[R4]. The previously created indexes were leveraged to examine the recommendations of the most helpful review for each product. Using a user score of 50 to separate generally positively and negatively reviewed products, it was shown that ~70% of top agree with the majority opinion of the product, which aligns with the aforementioned overall positivity of reviews. However, of those that go ‘against the grain’ (296), ~90% were negative reviews on a positively rated product. This shows that although most top reviews mimic the general sentiment, outliers are much more likely to be negative.

Summary

The patterns shown to exist for user reviews were deemed weak enough to settle concerns about the efficacy of user score as a standard measurement of sentiment. In any case, user score is already used as a measure of a product’s current quality by many.



Popular games retailer ‘Humble Bundle’ hosts a sale only for products with a high user score[lxvii].

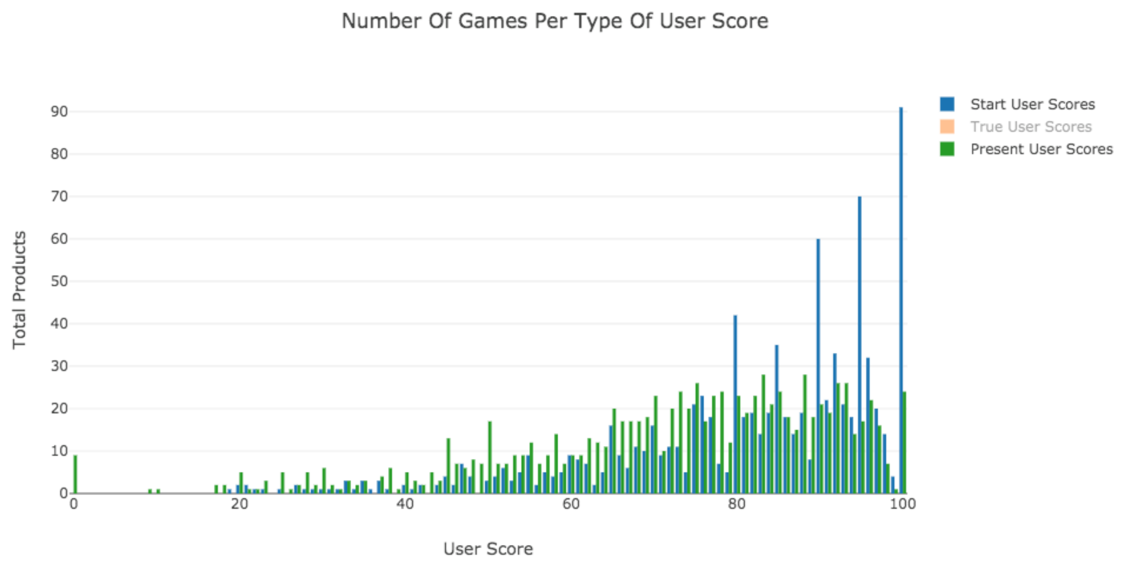
5.6 User Sentiment

Context & Purpose

Having studied the legitimacy of user score as a measure of sentiment, a large amount of the investigation was dedicated to exploring how and why this might change over time, seeking a link between these changes and developer engagement.

Overall Change

Firstly, the general change in user score over time was observed to be a negative one[R5]. While the majority of current scores (~75.81%) are still optimistic, they exhibit a greater spread into lower score values, and less exacerbated peaks at higher scores. This implies that reviews left by early adopters are more likely to be positive, and that sentiment wanes over time – many acceptable reasons could be suggested as to why, e.g. stakeholders self-reviewing, early optimism providing leniency.

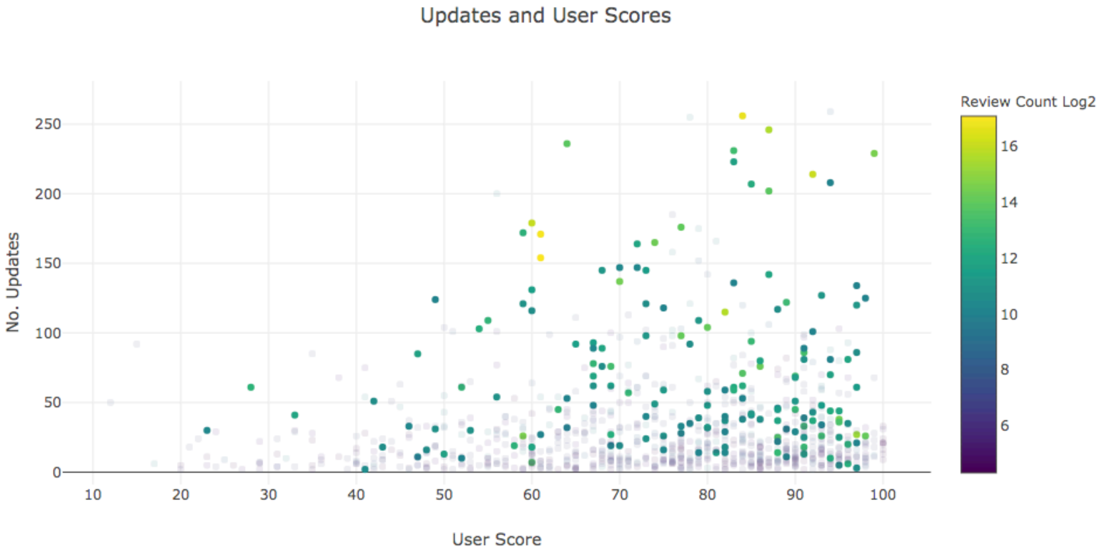


As the influence of a review's recommendation at each submission is *1/n+1*, user scores fluctuate more heavily at the beginning of a product's lifespan. Therefore, starting scores were computed at timestamps offset by a threshold of 20 reviews (whereby each following review has a less than 5% bearing on the overall score). This criterion reduced the candidate set to 897, as only products that had received at least one update at a point in time greater than receiving their twentieth review, and then furthermore having received at least one update after that point were eligible.

Product Updates

A developer will attempt to expand or improve the user experience of a product through updates; theoretically, this should encourage positive sentiment. Therefore, it was explored whether products that receive more updates also receive higher ratings[U1]. This route showed promise, but required much elaboration.

The mean of total updates per user score was found to only have a 0.56 correlation strength, which was not elucidative. To better visualise what was happening, a scatterplot was created to graph update counts against user score per game[U2]. Noise was removed from this graph by colour grading points against review count, as scores for products with more reviews are more apt representation of their quality. In this, it was seen that there is a slight trend towards a higher user score for products with more updates. When looking at a product as a static entity, however, it is not possible to understand whether these products simply have more updates because they entertain stronger support. Furthermore, the standard deviation of update totals increased with the score, showing that more updates are not strictly necessary for success.



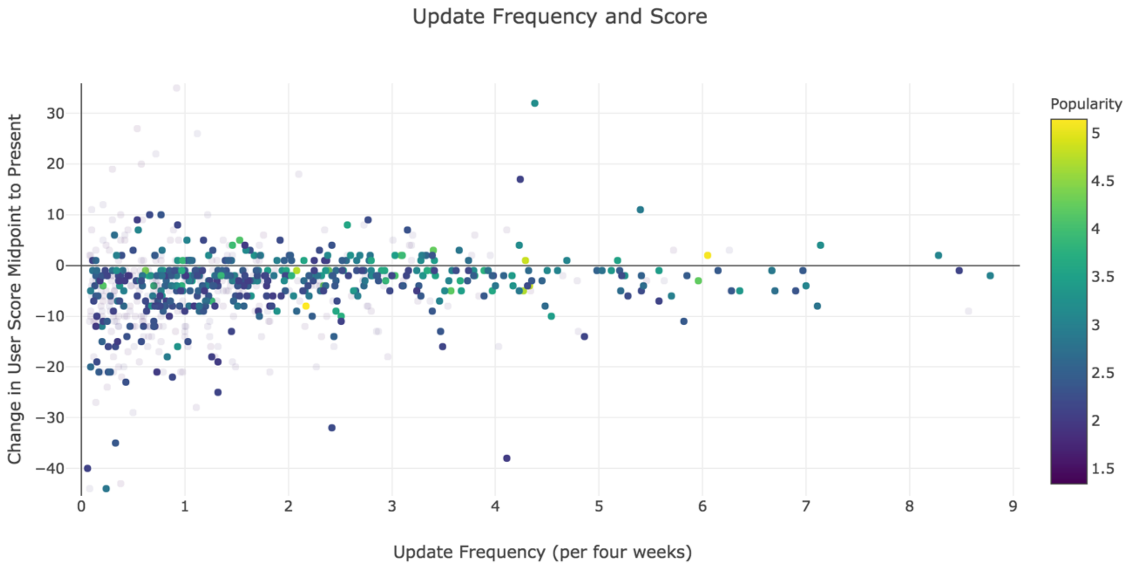
Product Updates Over Time

In order to study the effect over time, a more in-depth data representation was created for each product. This included the computed user score at the start and end points in their date range[U3].

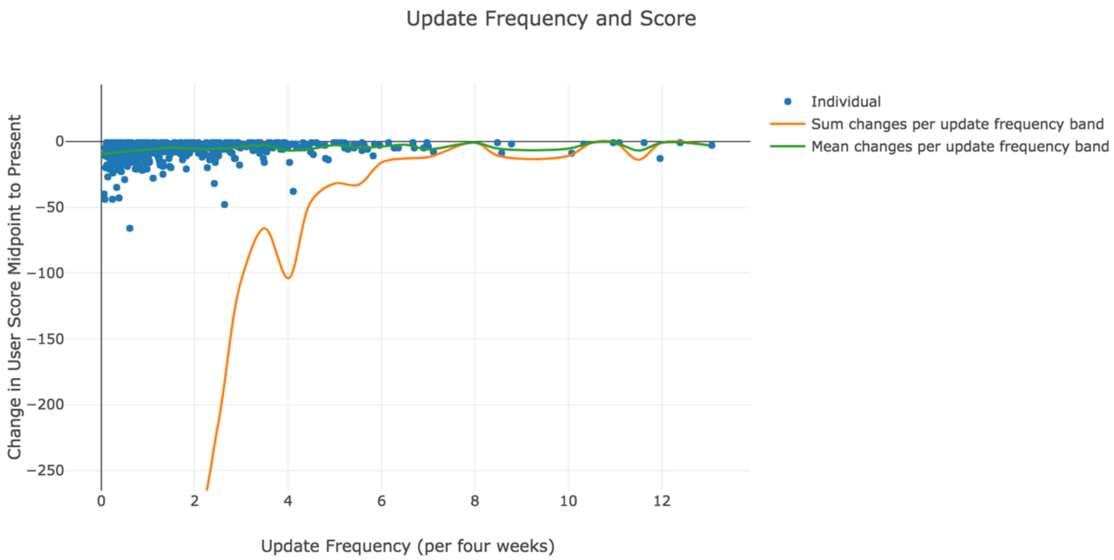
192 products (21.4%) increased from their initial score, while 58 (6.47%) remained the same, and 647 (72.13%) decreased. This number decreased was higher than expected, so user score at the mid-point of each product's life-cycle was also incorporated. 38 products were deemed to be too new for this further step (as their mid-points were too early). This actually promoted the previous results, showing that only 201 (23.4%) increased from their mid-point, with 127 (14.78%) remaining the same, and 531 (61.82%) decreased.

Several options were explored to determine why, and a number of statistics computed to aid this. The average absolute score increase for products whose score did increase was small, at just 2.37 points, with the maximum increase being 11 points. The highest score decrease was 28 by comparison. From their mid-point, it appears that most products do not deviate much at all; we refer to this as 'settling'.

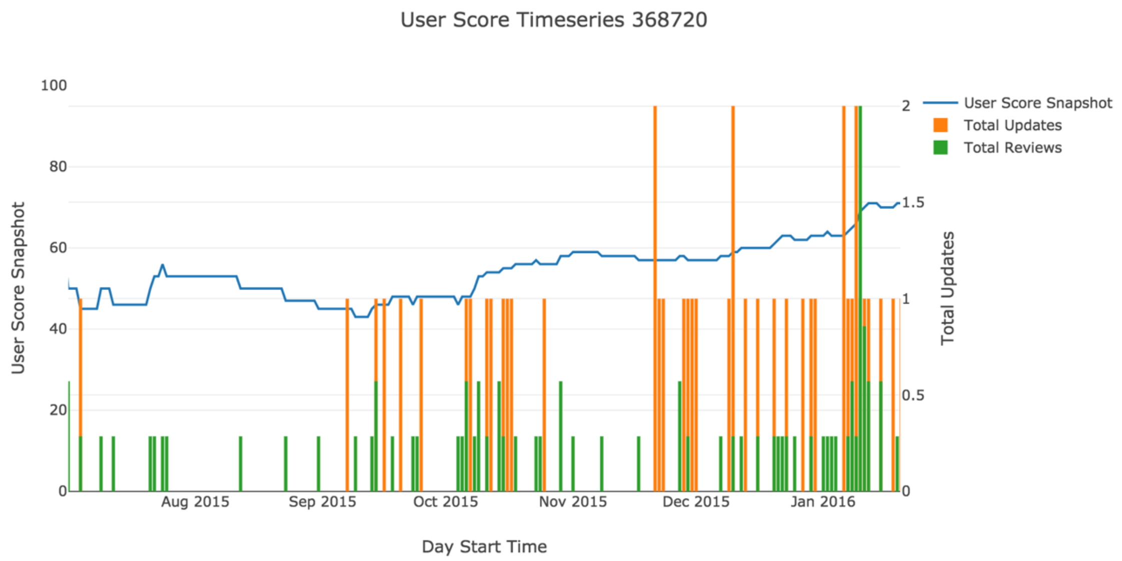
To factor in how updates were spread throughout the lifecycle, the update frequency per four weeks was computed for each product. Here, it was shown that the average update frequency of products whose scores increased was 2.15, as opposed to 1.85 for those whose scores decreased.



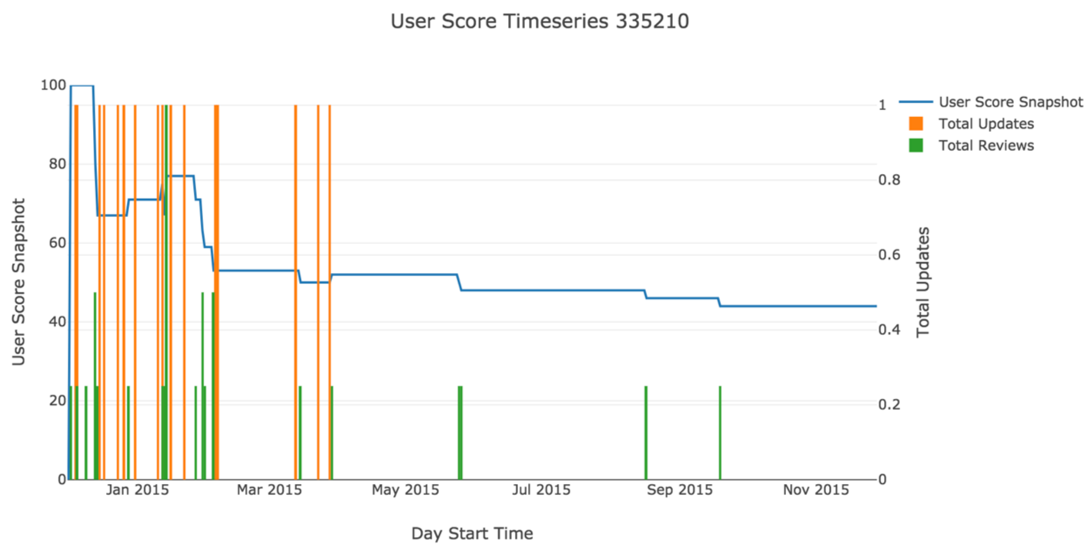
There was no correlation (0.15) observed across the whole set, however there was a greater density of of score drops for lower update frequencies, suggesting that more product updates can prevent drops in score. This can be seen by looking at the mean and sum of score changes for each update frequency.



Finally, to verify that updates uphold a score, a timeseries graph was constructed for different kinds of products, and correlation was sought between updates and positive changes in user score. The examples selected covered an increase, decrease and no change in score over time. Most notably, the product that increased over time only increased during bursts of updates (incurring more updates).

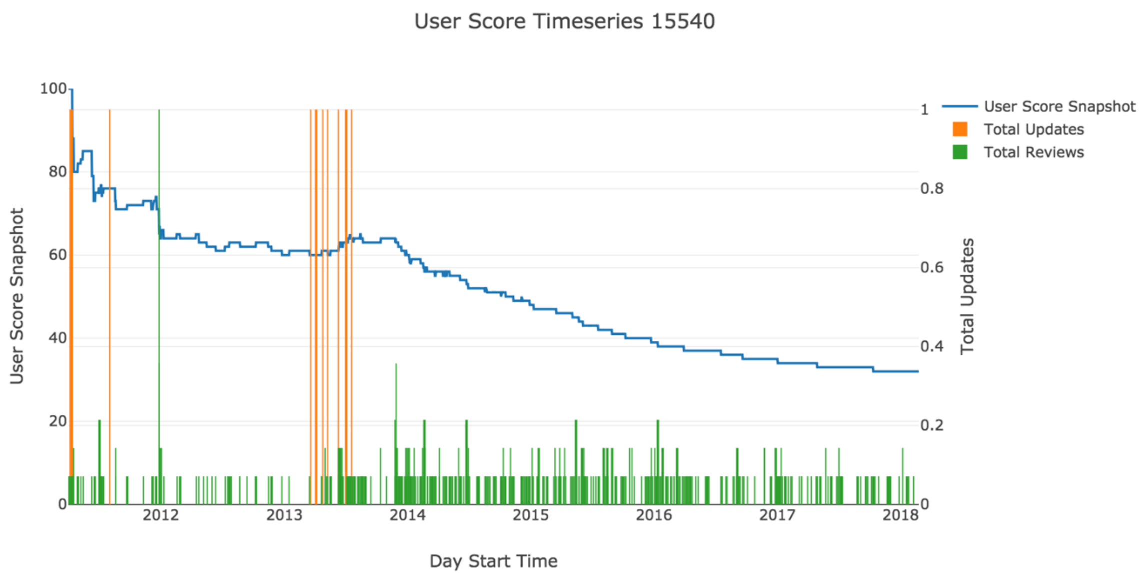


Conversely, the product that decreased over time did so during periods of inactivity.



Inactivity

It appears that although updates have a marginal positive impact on sentiment, periods of developer inactivity starkly correlate with large dips in user score. This is an observation detached from the content of an update, focusing instead on the time and propensity. Correlation is evident, but it is hard to gauge the impetus for a specific change without examining the context of the product at that point in time. The final piece of analysis in this section studied review contents anecdotally alongside timeseries graphs of products that appeared to have been abandoned either at intervals or entirely[U4]. They were found by scanning reviews for relevant key words such as 'abandoned', and filtering out products with high user scores and update frequencies.



Most of the graphs for these abandoned products looked similar to this.

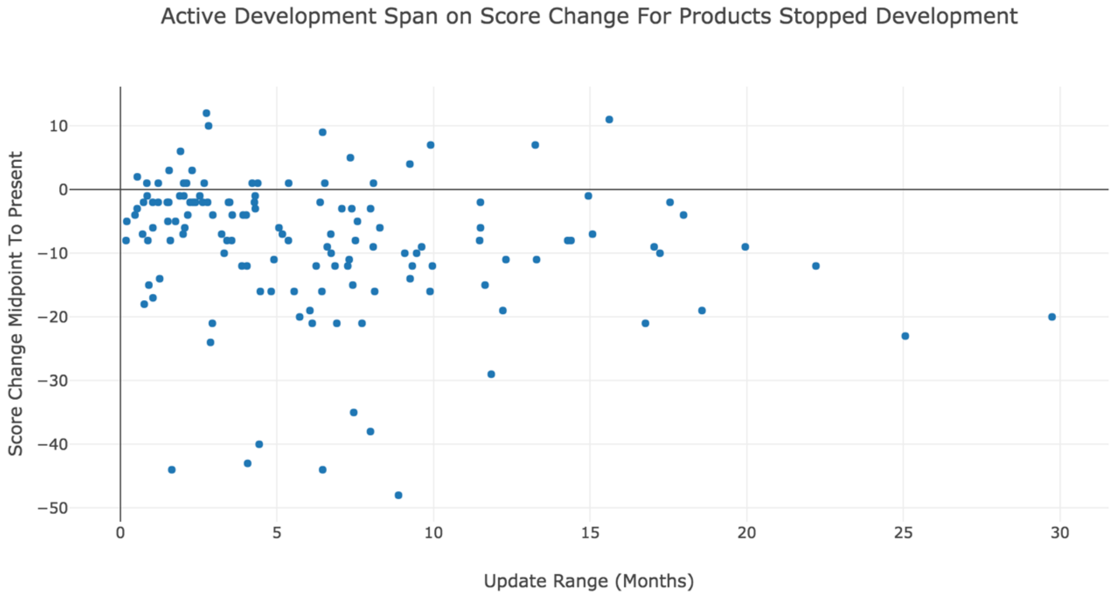
For such products, both positive and negative reviews shared similar sentiment:

"*I know that this is an Early Access game, but for Early Access to work, the devs need to put in some form of effort to actually develop the game further and get it out of Early Access.*" – Negative reviewer.

"*I wish if this game get [sic] more developed as soon as possible waiting for more updates.*" - Positive reviewer.

For each product that appeared to have stopped development, its 'story' – finished or abandoned, was able to be classified on a graph, using a combination of computed statistics related to the active development timespan. The number of products that had not been updated in at least a month prior to collection was 422, and the number that appeared to have stopped development altogether (no updates since their midpoint) was 142. This large figure is to be expected when considering that the concept of 'early access' was released 6 years prior.

Of those that stopped development, 84% decreased in score. The median score change of products that stopped development was -8.62 points, and those with shorter active development timespans were more prone to larger drops.



Summary

It is clear that the distribution of product updates influence user sentiment. As most products start with higher review scores, it seems some do not have much scope or room from improvement, and are quicker to plateau. However, in iteratively stepping down into the granularity of the problem, it is shown that early responsiveness and a consistent update frequency cultivates invested users and a higher settled score.

5.7 User Engagement

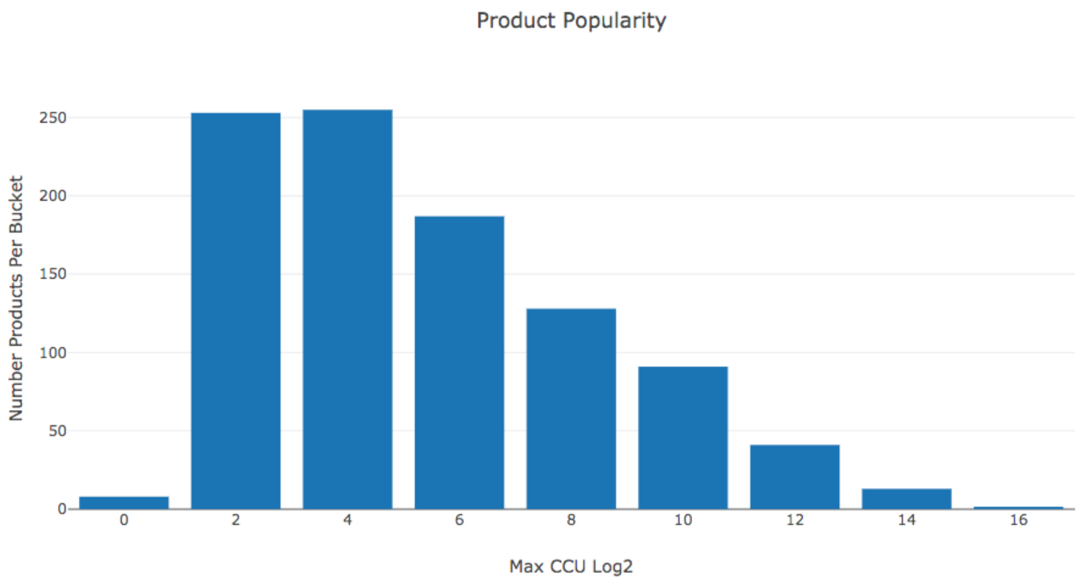
Context & Purpose

Having shown that product updates seem to affect user sentiment, it was possible to examine whether this translated into a transitive impact in user engagement.

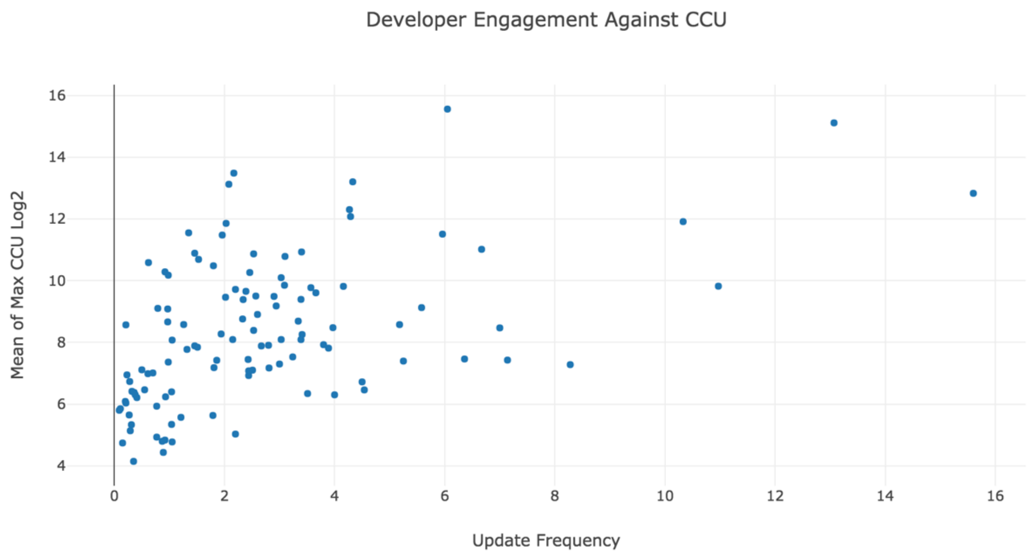
Updates & Popularity

It was first shown that products that have a higher update frequency have higher CCU[C1]. This was achieved by first making high-level observations of CCU across the entire set.

Each product has a max CCU – its peak popularity. The median of max CCU for all products is 54, and the mean is 1184. In visualisation, it was evident that few outlying products actually achieved a mass audience.

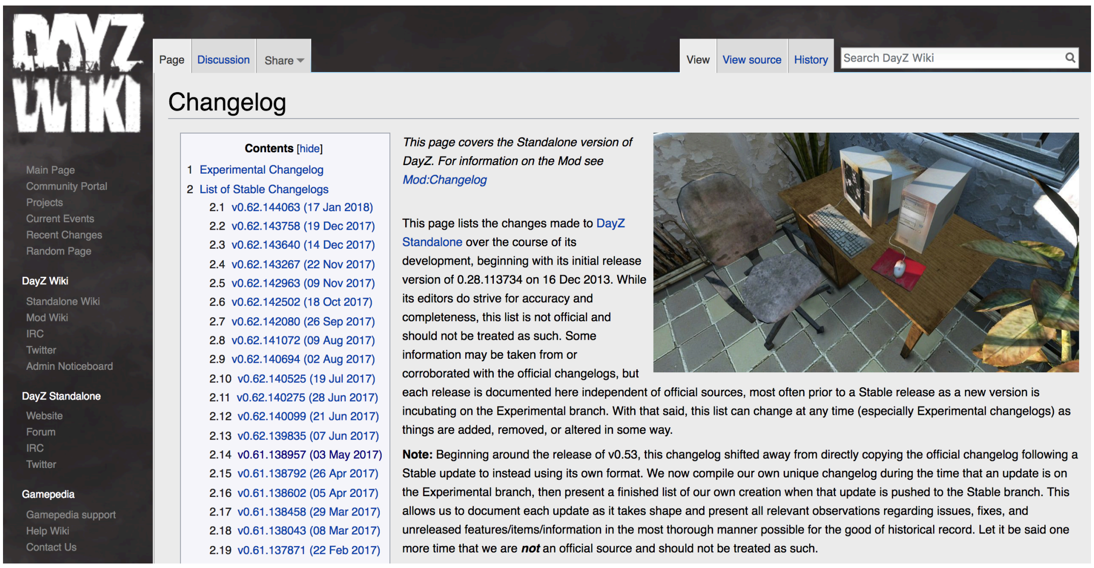


For these above-average products, those that had a higher update frequency achieved a higher max CCU.



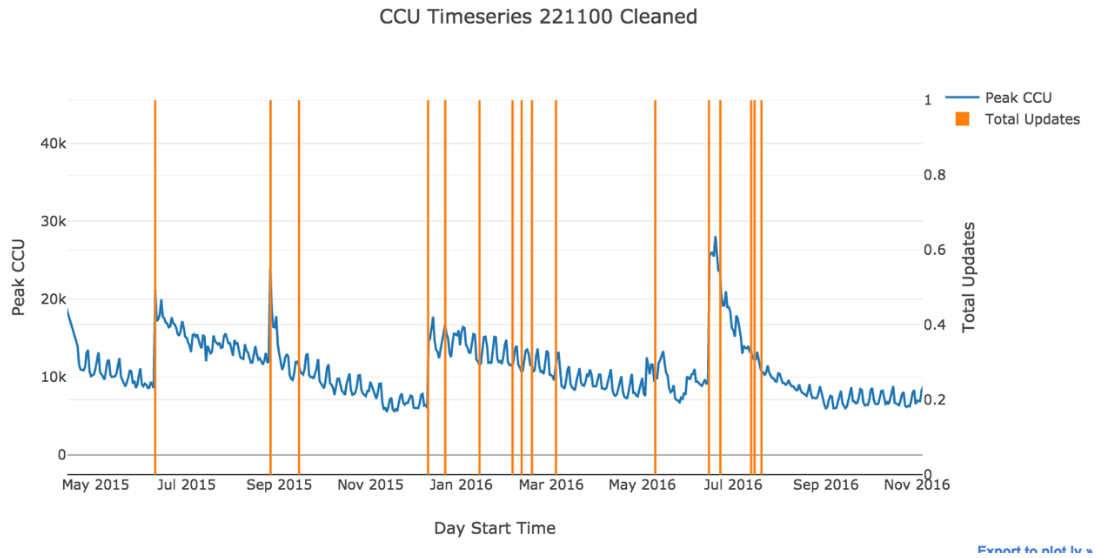
Quantifying Spikes

In attempting to assert whether updates were causing these greater peaks in CCU over time, the product with the most data in the set was selected. It was appropriate for investigation for many reasons. As an MMO and ergo ‘living’ game, its genre fosters updates as an incentive for recurring use, growing an array of tangible content and features over time. The game had not participated in any large promotions, reducing the influence of external factors. The wealth of data also resulted in the capturing of many interesting points, with a community curated list of meaningful updates helping inform clean-up.



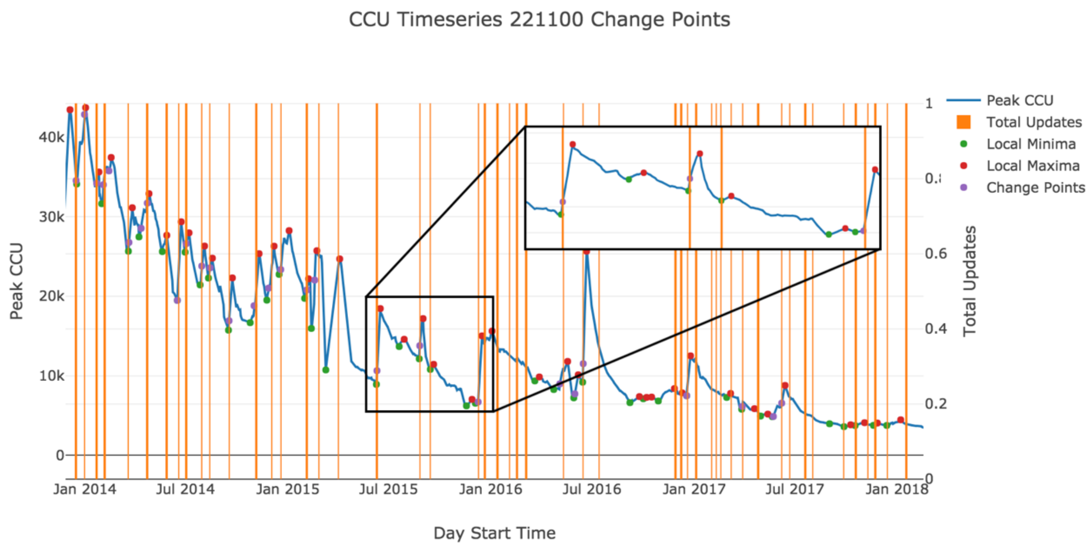
The product’s community changelog[lxviii].

The original signal displayed weekly seasonality, with local maxima occurring at weekends. This was converted to a stationary signal using a rolling mean of seven days. Furthermore, updates not included in the community-curated list were stripped, leaving only changes to in-game content.

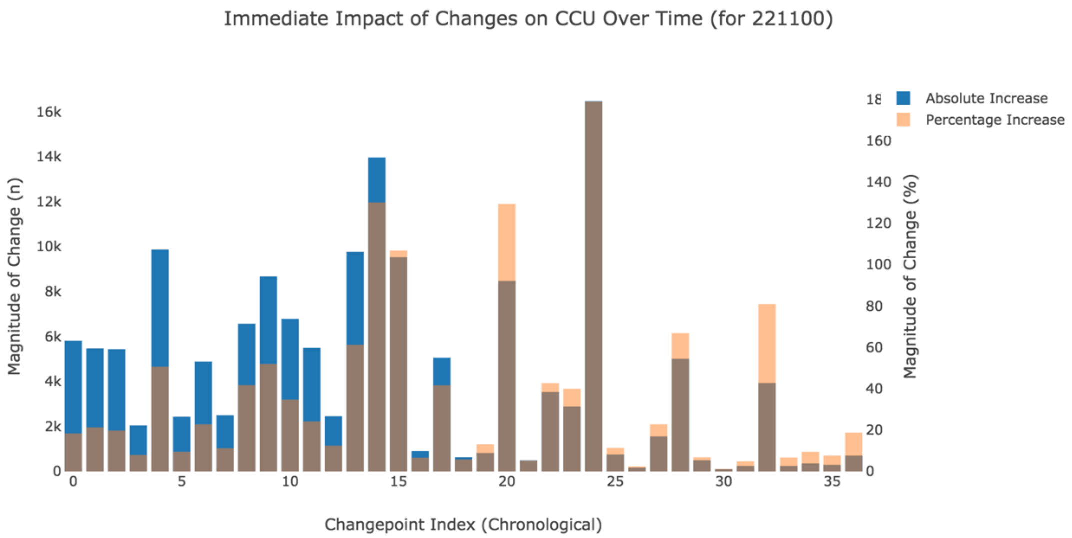


Weekly seasonality is apparent.

After formatting the data, it appeared at-a-glance that every visible positive change correlated with an update. In a complementary fashion, periods without updates appeared to result in steeper drops in CCU. To quantify this, changepoint detection was performed on the signal with a high threshold, capturing instances with a high rate of change. In every product, CCU undergoes an almost constant churn from the point of release. Resultantly, changepoints are observed at each significant local minima and maxima. The median distance of the 28 changepoints to both the nearest update and the nearest local minima was one day, a small enough distance to suggest a link between them.



Finally, the extent of each spike was measured using a function of the change in CCU between each local minima and maxima. The mean of these changes was 35.9%, and the median 19%. By graphing this change across time, it was observed that the potential for positive change increases over time (likely representing re-capturing lapsed users); this is demonstrated by large spikes. Additionally, the absolute impact of a change dwindles due to the churn of the user base.



The residual impact of each change was also measured by comparing the mean CCU of a time period prior and after. This showed that the lasting effect of each update was higher than expected, with the average change after one week being ~8%, and two weeks ~7%. While many did decrease sharply following, the maximum change after two weeks was 123%. Furthermore, comparing the greatest negative change in mean CCU (-11%), with the change for the same amount of time in a period with no updates, yielded a similar percentage decrease. This is encouraging, and shows that at best, updates can have a lasting positive impact that persists across time, and at worst do not hasten decline. This may suggest that updates improve retention, however whether the same users are being retained versus new ones would require further study.

Summary

It is clear that updates correlate with positive changepoints in CCU, and furthermore it was possible to quantify this for a product with a degree of manual investigation, fulfilling FO3.

6 Results & Evaluation

The findings from each investigatory step are categorised, and their resulting insights are grouped here for discussion. The entire project is then examined by an industry professional, and used to drive an evaluation.

6.1 Results

In the examination of user feedback:

- On average, 8% of reviews per product are submitted by users who have received the product for free, with software and casual game product genres receive the highest proportion of free copies (6% and 4% respectively)[R1].
- For the 78.2% of products with reviews left by users who received the product for free, those reviews were 5% more biased towards positivity[R1].
- For products that can be 'completed', the most 'useful' end ergo top reviews as voted by the community are left by users that have a logged time that correlates with the expected completion time for the product. For open-ended products, there is no strong correlation between the total usage and helpfulness of reviews, however the most useful reviews are left by users who have spent at least 6 hours using the product[R2, R3].
- Top reviews per product are slightly biased towards positivity in total, with 603 (~62%) of top reviews being positive, and 374 (~38%) being negative[R4].
- Although 70% of top reviews agree with the majority of other users having left reviews for that product, the outliers that disagree are significantly (~820%) more likely to be negative[R4].
- User sentiment is more positive at the beginning of a product's lifecycle, and almost product's user score wanes over time[R5].
- The majority (~75.8%) of all reviews are positive. The 'true user score' computed from this is more positive than the one reported by the platform, which implies that reviews left by users who acquired the product outside of the platform are more favourable[R5].

In examining the link between developer engagement and user sentiment:

- While a higher total of updates is not necessary to achieving a positive user score, products that have a larger number of updates logged receive more positive recommendations. All products with more than 125 updates have a user score higher than 50 (positive), and conversely, products scored negatively rarely reach more than 100 updates[U1, U2].
- Most products drop in user score over time. They all exhibit uncertainty at the start of their cycle, beginning high and then entering a period of fluctuation where updates are the most impactful. Most products do not largely deviate from the user score they have at the midpoint of their release cycle, and ~10% did not change at all from this point[U3].
- Product updates appear to be capable of increasing the user score of a product, with positive sentiment mostly achieved through clusters of updates[U3].

- Frequent product updates appear to uphold a user score, preventing it from dropping[U3].
- The average active development span for products is 16.1 months, and the optimal length appears to be between 3 and 15 months. Longer timeframes than this result in a drop in sentiment[U4].
- Periods without updates correlate with a drop in user sentiment[U4].
- ‘Abandonment’ and ‘completion’ can be classified using a combination of user score and updates over time[U4].

In examining the link between developer and user engagement:

- Most products remain relatively unseen, with the median for max peak daily CCU forever being 54[C1].
- There is a slight trend towards higher CCU for popular products with a greater update frequency versus popular products with a lower update frequency[C1].
- Product updates correlate with positive changepoints (spikes) in CCU, and demonstrate a lasting effect that appears to combat churn for a period of time. Quantifying this for a popular non-linear product sees an average uptick of ~35% increase in the current CCU per time of update. While this is fairly consistent over time, the absolute impact diminishes in tandem with churn[C2].

With a new understanding of some verified strategies that have promoted success factors in previous ‘early access’ products (and led to failures), developers may use the results above in order to plan their own. This may remove uncertainties in the process, e.g.

- The length of development time preceding a full release.
- A release schedule of updates throughout the lifecycle.
- Specific qualities of users to target for better feedback.

6.2 Industry Survey

An industry survey was conducted with the intention of evaluating critical aspects of the project including its justification from a business perspective, the methodology by which is was undertaken, and finally the applicability of its findings. This will furthermore be used to help assess to what degree the functional and personal objectives have been met, fulfilling FO4. The full transcript can be found in the appendix[9.2].

The survey was taken by Joel Graham, Director of Analytics and Data Science at Jagex, and feedback was largely positive on the 'Likert scale', with additional elaborative comments being both positive and constructive.

In assessing the project's motivation, Joel noted that there is a strong need for research in the area of NPD, which would in turn be very useful at his company. This is due to having shared little information about this process with others, in addition to "for the first time [having] a large dataset that could provide answers to some of these questions". He sees such work as necessarily cumbersome, likely requiring a "*commensurately large amount of resource*" [S1] in order to comprehensively review robust links to longer-term success, however also describes this as "*worth funding*".

In assessing the project's methodology, Joel relayed strong agreement as to its relevance to his own professional team. He agreed that the data collected has sufficient breadth, and stated that the technology choices are "*a perfect fit for how we work and the way that we document research for potential re-use by others*". The only substantial difference pertains to privacy requirements found in a corporate environment that are not necessary here. In regards to reproducibility, it was agreed that although he was confident his team could utilise the work to their benefit, a key addition that is lacking would be a programmatic record of operations undertaken during data 'munging' to transform it to the final 'groomed' set[S2].

In assessing the results of the investigation, Joel only slightly agrees that they align with his own expectations. He notes that there is value in both proving and disproving them: "*the findings are in many cases a validation of my pre-existing expectations, which is valuable both as a reassurance that our existing and less formal or objective domain knowledge that we have been using to make decisions is accurate, as well was providing a methodology to maintain an objective view on the continued truth of this*". Here, the idea of repeating the tests at time intervals to update these expectations was raised as a new consideration coincidentally achievable via the system produced. In ranking those that subverted his expectations as more useful, he placed great emphasis on the notion of "*using [those] findings to make fundamentally different decisions about the path of product development/release based on early metrics*"; these included things like reviewer bias and the impact of updates on scores and engagement.

Has also agreed that future investigation would be worthwhile. His suggestions were to first attain second opinions on the ETL process to "*reassure the replicability of it*", and then to continue to track the evolution of the problem space over time to promote its re-use. He also alluded to feature ranking in preparation for machine learning as a logical next step[S3].

6.3 Evaluation

The project architecture used to generate the findings has many qualities that make it useful in a number of circumstances. It is extensible due to its system architecture, and accessible through its deployment architecture. This is evident as it was able to be improved and reused during analysis where necessary, and furthermore shared remotely with industry sponsors. This is a testament to its versatility, and for a marginal server cost it was able to collect and process enough information, combined with its accompanying documentation[9.1] to fulfil both FO2 and FO5.

There are, however, notable limitations of the architecture that could be addressed in future. Most influentially it does not have the capability of live data streaming and ergo does not react to new data without manual effort. Upon reflection, this does not adequately reflect the fast-paced nature of the industry in question. Furthermore, some operations used to transform the data are not detailed in documentation. This was foregone for brevity, however in retrospect represents a barrier to reproduction. Finally, although the system was performant in regards to the project's specific needs, the lack of true multi-threading in message handling raises concerns when considering scaling the system to consume all available products (over 700 million) (as opposed to a subset of fewer than 2,000). It is likely that more work would have to be undertaken to accommodate these features.

The strengths of the findings themselves lie in their novelty. The dataset itself is constructed from component parts that do not appear to have been publicised together before, in an industry where data itself is a commodity used for many bespoke and private purposes. While some generalisations can be attained through public interfaces, the findings highlight the potential for combining multiple relevant sources to achieve a specific goal. In this case, they manage to consider the interplay between developer and user feedback as a reactionary measure recorded over time, with the accumulation of these metrics informing the success of a product. This has resulted in observations that represent 'stories' on a scale of positive to negative, and in these classifications of a product's ultimate journey towards completion, reasoning has been made against their failures and successes to be avoided and replicated respectively. In this way, the results show that the impact of developer engagement is quantifiable against user metrics, and that the metrics themselves have concrete biases that are explained. This can in turn be used to reduce uncertainty during NPD; for each discrete finding, its practical application is easy to envision. Each has its own place within the lifecycle to help maximise success factors outlined during research, and therefore can be used to inform the overall strategy, achieving the most substantial objective FO3.

The limitations of the findings are largely rooted in their specificity. Strong trends are only seen when looking at the minutiae of the overall problem, and are otherwise muddied during attempts to generalise. This is largely due to the fact that products that are 'creative' are inextricable from their context, and there are many more considerations that could be integrated into the system to aid the search for meaningful conclusions. This is exemplified in the number of known factors shown to be influencing the data. The architecture and dataset itself therefore hold further potential application[7.2] which may reduce subjectivity, however this would require a substantial amount of work. Additionally, some antecedent findings were already discernable from simpler analyses that have been shared historically; although these were necessary stepping stones towards the greater objective, they took a substantial amount of time to assert (e.g. legitimacy of user reviews). Ultimately, where the analysis benefits from its domain knowledge, it also suffers in places from a lack of mathematical depth that leaves some insights closer to estimation than fact.

7 Conclusion

7.1 Discussion

This paper's research manages to marry the more typically business oriented aspects of project management in technological industries with the development they depend on, emphasising their inextricable link. It benefits from the previous studies and expert opinion used to inform its investigation by identifying barriers to a product's success that a homogenous group of interested parties suffer from during the predominant and disruptive 'early access' release strategy, and furthermore derives observable criteria for study in order to explore these problems. To this end, FO1 was fully realised.

The system underpinning the project's data mining is unique insomuch as it combines a set of metrics in a manner hitherto unpublicised, and is therefore beneficial in an industrial context; its accessibility encourages further analysis and exploitation, and builds on a precedent of using the intricacies of competing products within a common platform to maximise success. Although improvements can be made to increase its robustness, the quantity of data processed was more than sufficient, achieving FO2.

In its findings, the paper shows that crucial aspects of the engagement cycle can be quantified, with the caveat of many circumstantial flags. Although each route would benefit from further investigation, by ambitiously exploring so many different relationships within the data, more confident reasoning behind each has been conveyed, thus satisfying FO3. Through rigorous evaluation capturing FO4, it is apparent that while many previous works fail at attempting to meaningfully predict product success, this paper succeeds in the promotion the adoption of successful methodologies *during* the NPD lifecycle. This aligns with the expectations of sponsors, who, through a networked visualisation of data, certify that some findings can be put to immediate use, thus also achieving FO5.

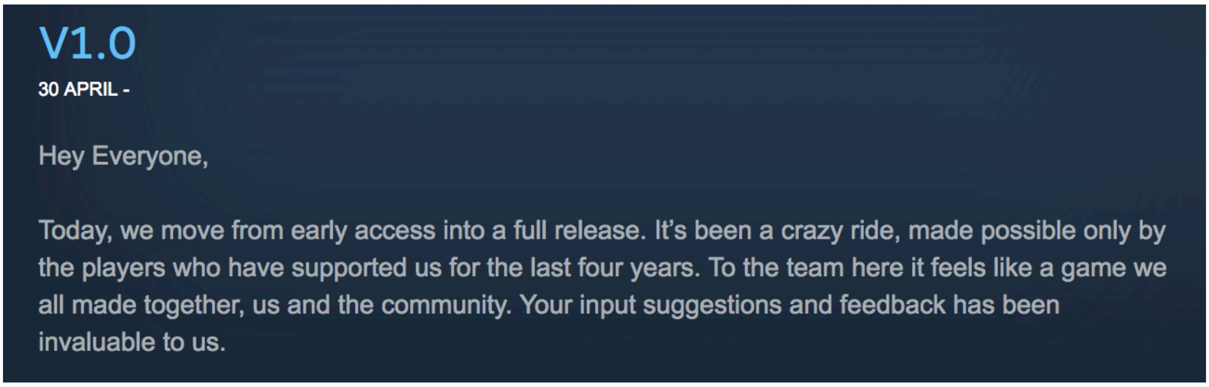
Ultimately, there are circumstances in which a product or team may not reap any benefit from engaging in a cycle of community-driven iterative development; the success of updates in achieving the effects documented is a reflection on the developer's ability to respond to the existence of useful feedback in a manner that garners community support. However, this is shown to not only be very achievable, but in most cases, essential. This sheds light on many previously subjective assumptions, laying the groundwork for further future study.

7.2 Future Work

There were some potentially useful avenues of exploration discovered during investigation that were not in the remit of this paper. Most notably, social change as an externally influencing factor on data was made apparent by otherwise unexplained spikes in the data. Sources online have already begun to showcase this in response to the rising prevalence of ‘community influencers’ and competitive play[lxix], which, if processed somehow, could make it a third category of engagement to augment this paper, and therefore another potential opportunity to promote a product’s success. Another useful metric would have been the intended ‘completion’ time of products. Unfortunately, there is a lack of public information at present regarding this, which condemns its incorporation.

A popular ‘streamer’ breaks records, with 628,000 concurrent viewers watching an ‘early access’ game.

The dataset also holds more potential for organic study. One example would be to place greater emphasis on comparing software products with video games, however this was not done here, as software products represented insufficient data for meaningful analysis. Another potential comparative study could focus on products that have graduated from ‘early access’ status.


V1.0

30 APRIL -

Hey Everyone,

Today, we move from early access into a full release. It's been a crazy ride, made possible only by the players who have supported us for the last four years. To the team here it feels like a game we all made together, us and the community. Your input suggestions and feedback has been invaluable to us.

A community update by a developer whose game had just left early access.

The deliverable material already provided also forms a basis for more desirable work. Most notably, it provides a set of features and proven relationships of the analytics for which there is much potential for predictive modelling. An important omission from this paper is the content of updates. Using a combination of file sizes, and the apparent impact of an update on sentiment and CCU, the impact of an update could be classified using machine learning. Furthermore, a feature ranking should be undertaken in order to model the impact of updates, which would be a perfect candidate problem for a supervised neural network. This could convert the advice listed in findings into a truly predictive and transportable tool for use during development, removing the need for manual comparison, which is a desirable if far-reaching concept. This would automate the process of producing predictive analytics.

8 References

[i] Pandit, P. (2015). IoT analytics brings new levels of innovation to new product development. Retrieved from https://www.networkworld.com/article/2947401/big-data-business-intelligence/iot-analytics-brings-new-levels-of-innovation-to-new-product-development.html
[ii] Kettunen, P. (2009). Agile Software Development in Large-Scale New Product Development Organization, 43-45
[iii] Goodfellow, I. and Bengio, Y. (2016) Deep Learning, 15
[iv] Kroll, A. (2017). Steam Blog - User Reviews Announcement. Steam Community. Retrieved from https://steamcommunity.com/games/593110/announcements/detail/1448326897426987372
[v] Galyonkin (2016). Games on Steam by release year. [image] Available at: https://twitter.com/Steam_Spy/status/804072335997358084/photo/1?tfw_creator=gamasutra&tfw_site=gamasutra&ref_src=twsrc%5Etfw&ref_url=https%3A%2F%2Fwww.gamasutra.com%2Fview%2Fnews%2F286644%2FSteam_Spy_Over_a_third_of_all_Steam_games_were_released_this_year.php
[vi] Sandqvist, U. (2015) The Games They Are A Changin', 12-16
[vii] Brechner, E. and Waletzky, J. (2016). Agile project management with Kanban.
[viii] González-Piñero, M. (2017). Redefining The Value Chain Of The Video Games Industry.
[ix] Keith. (2010). Agile Game Development with Scrum. Pearson India.
[x] Clinton, K. (2010). The State of Agile in the Game Industry. [online] Gamasutra. Available at: https://www.gamasutra.com/view/feature/4295/the_state_of_agile_in_the_game_.php?print=1
[xi] Keith. (2010). Agile Game Development with Scrum. Pearson India.
[xii] Job, J. (2015). Scrum Diagram. [image] Available at: https://jordanjob.me/2015/12/07/scrum-diagram/
[xiii] Perspectives, T. (2016). Fail Fast: The Second Ingredient For A Successful Company. [online] Forbes. Available at: https://www.forbes.com/sites/teradata/2016/03/02/fail-fast-the-second-ingredient-for-a-successful-company/#fe317a274a71
[xiv] González-Piñero, M. (2017). Redefining The Value Chain Of The Video Games Industry.
[xv] Chakravorti, B. (2004). The New Rules for Bringing Innovations to Market. Harvard Business Review, (March).
[xvi] Perspectives, T. (2016). Fail Fast: The Second Ingredient For A Successful Company. [online] Forbes. Available at: https://www.forbes.com/sites/teradata/2016/03/02/fail-fast-the-second-ingredient-for-a-successful-company/#fe317a274a71
[xvii] Drabløs, C. (2018). What influences crowdfunding campaign success. University Of Agder Department of Economics and Business Administration.
[xviii] Engelstätter, B. and Ward, M. (2016). Strategic timing of entry: evidence from video games. Journal of Cultural Economics, 42(1), pp.1-22.
[xix] Tassi, P. (2014). Why It's Scary When 0.15% Mobile Gamers Bring In 50% Of The Revenue. [online] Forbes. Available at:

https://www.forbes.com/sites/insertcoin/2014/03/01/why-its-scary-when-0-15-mobile-gamers-bring-in-50-of-the-revenue
[xx] ITJobsWatch. (2018). Games Tester Jobs. [online] Available at: https://www.itjobswatch.co.uk/jobs/uk/games%20tester.do
[xxi] Zimmerman, E. (2003). Play as Research: The Iterative Design Process.
[xxii] Zimmerman, E. (2003). Play as Research: The Iterative Design Process.
[xxiii] Phan, M., Keebler, J. and Chaparro, B. (2016). The Development and Validation of the Game User Experience Satisfaction Scale (GUESS). Human Factors: The Journal of the Human Factors and Ergonomics Society, 58(8), pp.1217-1247.
[xxiv] Player Research (2018). Playtesting Methodology. [image] Available at: http://www.playerresearch.com/
[xxv] González-Piñero, M. (2017). Redefining The Value Chain Of The Video Games Industry.
[xxvi] González-Piñero, M. (2017). Redefining The Value Chain Of The Video Games Industry.
[xxvii] González-Piñero, M. (2017). Redefining The Value Chain Of The Video Games Industry.
[xxviii] Symborski, C., Barton, M., Quinn, M., Korris, J., Kassam, K. and Morewedge, C. (2016). The Design and Development of Serious Games Using Iterative Evaluation. Games and Culture, 12(3), pp.252-268.
[xxix] Florén, H., Frishammar, J., Parida, V. and Wincent, J. (2017). Critical success factors in early new product development: a review and a conceptual model. International Entrepreneurship and Management Journal.
[xxx] Cook, A. (2018). Four issues shaping the future of video game marketing. [online] Campaignlive.co.uk. Available at: https://www.campaignlive.co.uk/article/four-issues-shaping-future-video-game-marketing/1221970
[xxxi] Cohn, M. (2015). Succeeding with Agile. Pearson.
[xxxii] Steinberg, S. (2011). Video Game Marketing: The New Bible Part 1. [online] GamesIndustry.biz. Available at: https://www.gamesindustry.biz/articles/2011-05-09-video-game-marketing-the-new-bible-article
[xxxiii] Jawecki, G. and Fuller, J. (2008). How to use the innovative potential of online communities? Netnography - an unobtrusive research method to absorb the knowledge and creativity of online communities. International Journal of Business Process Integration and Management, 3(4), p.248.
[xxxiv] G.Zoeller. "Aggregated spatial visualization", Image, available at: http://www.gdcvault.com/play/1012227/Development-Telemetry-in-Video-Games
[xxxv] Galankis, R. (2014). Agile Game Development is Hard
[xxxvi] Game Developers Conference. (2018). Call For Publications. [online] Available at: http://www.gdconf.com/conference/c4p/
[xxxvii] Powell, R. (2016). Positive and Negative effects of Game Analytics in the Game Design process.
[xxxviii] Powell, R. (2016). Positive and Negative effects of Game Analytics in the Game Design process.
[xxxix] Oldschool RuneScape Community Polls. (2018). [online] Available at: http://services.runescape.com/m=poll/oldschool/index.ws
[xl] Sentiment Classification using Machine Learning Techniques. (2016). International Journal of Science and Research (IJSR), 5(4), pp.819-821.
[xli] McGarry, K. and McDonald, S. (2017). Computational methods for text mining user posts on a popular gaming forum for identifying user experience issues.

[xlii] Vladimiro, R. (2015). The Player Lifecycle. [image] Available at: https://ongamesndata.wordpress.com/2015/07/10/the-player-lifecycle/
[xliii] Kim, S., Choi, D., Lee, E. and Rhee, W. (2017). Churn prediction of mobile and online casual games using play log data. PLOS ONE, 12(7), p.e0180735.
[xliv] Orland, K. (2014). Steam Gauge: Do strong reviews lead to stronger sales on Steam?. [online] Ars Technica. Available at: https://arstechnica.com/gaming/2014/04/steam-gauge-do-strong-reviews-lead-to-stronger-sales-on-steam/
[xlv] Champandard, A. (2015). Deep Gabe: Machine Learning from Steam Data.
[xlvi] Lancaster University (2012). Changepoints in homogenous segments. [image]
[xlvii] Ahuja, A. (2015). Changepoint Detection
[xlviii] Cowpertwait, P. and Metcalfe, A. (2009). Introductory Time Series with R. New York, NY: Springer.
[xlix] Brownlee, J. (2017). How to Remove Trends and Seasonality with a Difference Transform in Python. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/
[l] Kourentzes, N. (2017). Benchmarking Facebook's Prophet. [online] Kourentzes.com. Available at: http://kourentzes.com/forecasting/2017/07/29/benchmarking-facebooks-prophet/
[li] GameRefinery. (2018). [online] Available at: http://www.gamerefinery.com/
[lii] Devtodev.com. (2018). A full-cycle analytics solution for game developers. [online] Available at: https://www.devtodev.com/
[liii] Steam Store. (2018). [online] Available at: https://store.steampowered.com/
[liv] Usher, W. (2018). New Steam Update Means The End Of Steam Spy. [online] Cinema Blend. Available at: https://www.cinemablend.com/games/2402601/new-steam-update-means-the-end-of-steam-spy
[lv] Steam Web API Documentation (2014). [online] Available at: https://partner.steamgames.com/doc/webapi
[lvi] Steam DataBase (2018). [online] Available at: https://steamdb.com/
[lvii] Steam Spy (2018). [online] Available at: https://steamspy.com/
[lviii] Steam Supported Currencies (2018). [online] Available at: https://steamcommunity.com/dev/apiterms
[lix] UK Government (1998). Data Protection Act 1998.
[lx] Kaggle. (2017). The State of ML and Data Science 2017. [online] Available at: https://www.kaggle.com/surveys/2017
[lxi] Franusic, J. (2016). Thoughts after 6 months of literate programming. [online] Github. Available at: https://gist.github.com/jpf/d71453f535065a0d9281672152541386
[lxii] Anorov (2015). cloudflare-scrape.
[lxiii] Curtis, T. (2012). Steam sales: How deep discounts really affect your games. [online] Gamasutra.com. Available at: https://www.gamasutra.com/view/news/174587/Steam_sales_How_deep_discounts_really_affect_your_games.php
[lxiv] Derdenger, T. (2014). Technological tying and the intensity of price competition: An empirical analysis of the video game industry. Quantitative Marketing and Economics, 12(2), pp.127-165.

[lxv] Lindbergh, B. (2016). Length Matters. [online] The Ringer. Available at: https://www.theringer.com/2016/8/25/16038806/video-game-length-playtimes-f7b8e38f949f [Accessed 4 May 2018].
[lxvi] How Long To Beat – Statistics (2018). [online] Available at: https://howlongtobeat.com/stats.php
[lxvii] Hillier, B. (2017). Humble Bundle 'Overwhelmingly Positive' Sale. [image].
[lxviii] DayZ Wiki. (2018). Changelog. [online] Available at: https://dayz.gamepedia.com/Changelog [Accessed 4 May 2018].
[lxix] Business Insider (2018). Esports Popularity Forecast. [image].

9 Appendix

9.1 Guidance for Reproduction

The project's source code can be found at the link below, along with a 'wiki' that summaries the key sections of this paper, up-to-date online references, system requirements, and, most importantly, steps for reproducing the dataset used for investigation. A downloadable database snapshot is also provided.

https://www.github.com/joshhills/dissertation-project

9.2 Industry Interview Transcript

Below is a full transcript of the interview with Erik Lagel used to verify background research.

<u>Disclaimer</u>

The timespan of this interview may be brief owing to circumstance - it will be logged in post.

<u>Intention</u>

This intention of this interview is to help inform the applicability of the project, and furthermore provide context and research; this is reflected in the questions/talking points, which seek to verify of dispel assumptions I have about those areas.

<u>Meta</u>

The participant will have the aims of the project described to them.

Name: Erik Lagel

I1 Is there a formality to choosing a new content to develop? Who talks to who?
I2 Is the development of new content timeboxed? How do timescales factor into what you do?
I3 Is the development of new content community sourced? How does community feed into what you do?
I4 To what degree, if any, do you think this varies per-organisation?
I5 Do you look to other organisations than your own for direction?
I6 What are the main problems you face trying to develop content that deviates from the norm? I'm thinking more problems with process as opposed to existing fans.
I7 Would you describe what you do as data driven? Would you prefer to make decisions based on data?

Duration: 0:31:10

Josh: I wasn't sure, but it's recording now, I think. So, Erik, the first one I got because I thought it might lead into more things is, is there a formality, at least here, to choosing new content to develop? Who talks to who and what is the process like for making a new thing entirely? I know you've got NPD stuff, which is quite new.

Erik: The answer is yes. There it is.

Josh: So you've got a pipeline?

Erik: We've got a pipeline, exactly, a pipeline that has currently 12 steps. It goes from preselection to assessment, as well from production and business up to a final deal that would be signed by the end of the process. So it's been designed in coordination with marketing strategy and development to ensure that every step makes as much sense as possible. There's more detail about that time now, but basically we try to - do you want me to go through it or not?

Josh: If you wouldn't mind, just as high level.

Erik: High level. So, discovery is basically business development. It's the activity of trying to source new games, depending on whether they are purchased; whether we look for something. Basically, have a list of a number of dozens and dozens of games that might be of interest. Skimming is a small number of people trying to figure out, is this actually a living game for Jagex? Is this worth checking because the game doesn't seem completely crap? So basically what we're saying, it has to be more than a two out of five.

If it's more two out of five, I get involved and some more people, some handpicked, cherry-picked people in the company to play it and give a more, I would say, professional assessment on whether it's worth continuing further or not. At each stage, of course, any game project can drop and be eliminated because it doesn't fit the criteria.

If we have the conviction that, hey, we're on to something here, it branches out in terms of, we need to create a live business case. That actually allows us to understand whether, based on the study of the game, it has the potential to actually create a great case. And we have a full test through the company where we recruit a number of players in the company to play test a game and give us their opinion about it; whether they feel it's actually a fun game. We try to actually address the fact that neither of the people in this can cover everything.

For each test stage we're already starting to find - we can start to develop or try to figure out what they are up to. We're interested in your game because we've played it and we figured out that the business case seems okay from a distance. So we try to understand what kind of relationship we could have with them, figure out if they are serious, if their studios are okay and so on.

And then we've got what we call validation test, which is basically going to more external. And whether we will do it or not depends on the project, but external validation with, I would say, head player research or VGM that actually allows us to figure if these are really external players with no bias that we could have Jagex. Because we are all gamers to some degree, so we want to actually play test this, the results of which come into the full business case.

The full business case will tell us, okay, that's basically preparing investment. Should Jagex invest into title? Does Jagex think it's actually worth that amount of money, because it has the potential to reap these benefits? And then, at the same time, we start negotiating a contract with the partner, up to a point where the due diligence is operated in the same process where we figure out, in terms of verity, in terms of finance, in terms of legal, whether they fit all of the criteria. And we gather all of this information in one presentation for the executive team and the green light, which leads, hopefully, if it's successful, to signature of a deal.

This is the birth of a new project. There it is.

Josh: Right. So it's like a big filtration process that hopefully-

Erik: It definitely is a pipeline, yes. It's a filtration process. It's a funnel. So you had another slide, which I haven't printed, but basically you would probably have 200 opportunities per year to filter down to one or two projects per year.

Josh: Right. Okay, awesome.

Erik: I hope it answered your question

Josh: It did, definitely. I find that interesting too because I think in a lot of cases it's like magic to a lot of people and you miss a lot of the steps that go into it. It's seems to be like a lot of actual communication with real people. Does any of that look like the outside of the company in terms of fans or player bases or other player bases? Do we ever ask anybody else what they think or is it all kept internal until we know that we want to develop something?

Erik: What do you mean else, as in who else? Yes, the validation test, as we call it, step 7 is exactly that. It's to actually ask people outside of our company under a DA; I'm trying to make sure that we've got an independent assessment of whether the game is actually worth going further or not. Because obviously it's a research more than asking for feedback we certainly wouldn't ask external people, "Is this game going to make money?"

Josh: Fantastic. You've already given me an inkling. I think the bit below that has timescales and things like that, but just for the sake of the question. Is the development of new things, like how do timescales factor into what you do? If it is a big game does it get more time, a little game…?

Erik: Yes, so it's basically how long is a piece of string. We used to have a much stricter time line in the first version of this process, which basically we broke. Because from our publishing VP experience, basically it could last 12 weeks, it could last 30 weeks. It could last half a year, a year, depending on what kind of games you're talking about and whether the negotiations and the progress goes well.

So the shortest seems to be three months. Three to six months is the average, but it could last longer and potentially be even shorter if we are good.

Josh: Do you think that depends on the company or the team rather?

Erik: It depends on everything I would say. It depends on the game itself. Is it a small game? If it were some kind of casual game it's easy to assess, it depends then on the risk. Is there a lot of money involved? Is there a lot of benefit? If there's a lot of millions we need to invest, it's going to take longer. Is the relationship going well? If we've got people who actually were on the same line all the time it's going to be easy to discuss and get further. Does the business case seem clear and does it make sense? That's a no-brainer or is the risk bigger and so on. All of this; impact how long it takes to actually get to the information we need.

Josh: Obviously, I can see all the spreadsheets on your screen so I think I know the answer to this one, but would you describe what you do as data driven? Would you prefer to make decisions based on data or is it more about heuristic communication?

Erik: It's very interesting because yes, that's part of it. And funnily enough we had Joel, who you have met, I guess, who was presenting yesterday about data driven. And he repeated endlessly, which I agree 100% with, that data driven is difficult. Or you should step away from data driven because the data shouldn't guide your decisions. It's still a creative industry. It's still something that actually is part magic, part science, which is why we are supposed to be scientifically creative and not just scientific. We are not working in banks. We're not working in industry. We are working in an art field.

So to answer your question, part of it has to be supported by data but data driven wouldn't be as much as data supported. It's the fact that we are still creators. We are still trying to make a game that should appeal to the subconscious of people that figure out that actually, that's a great game. That people want to play and that you can't really measure easily. The only thing you can measure is the results but the intent is difficult.

So I'd say yes, there's a lot of data involved, as you can see, but it's not the end of it all. It has to be driven by belief that there's a potential.

Josh: So you're saying if you were to rely on the data too heavily then you might stifle that creativity?

Erik: Exactly that. I was having a discussion earlier this morning, the fact that if you asked me, since 90% of the games do fail and only 10% succeed, probably we shouldn't do any games because the risk profile doesn't work. So at one stage you need to shoot for the 10% and bet something on the fact that actually, it's got a chance to be in that 10%-

Josh: And the data is used, I guess, to just minimise the amount of risk in that process and make sure that the ones that you are choosing are the ones with the most potential?

Erik: Yes, exactly. And be always aware that there’s a very big percent of chance and luck in that process that we try to minimise as much as we can. The more work you do and more thorough you are, the less chance or luck plays a role but still it will be around that. And the only think to figure out is to measure the results. And that’s why early access probably makes sense because you’re go into early access to figure out if it’s worth spending marketing dollars to make sure that people come to the game. Because if people in early access find it’s not really interesting, then it’s probably a waste of money doing that.

Josh: So do you think there’s value in looking at what other people are doing? Do you feel a lot of what you do is studying what worked or didn’t work for other people and then trying to see if you cross apply it?

Erik: That’s why we put limits. It’s coming back to the data-driven thing. Again, the discussion I had this morning is the fact that forecasts essentially are trying to predict the future. And whoever had done the forecast of all of these games, if you look at Team Fortress 2, if you look at The Division, if you look at Wildfire, no one would have modelled this. No one would have predicted this. Every forecast model on all of these real numbers would have been false.

Josh: Do you think that’s just because that’s a bit too reductive?

Erik: In which way?

Josh: When I was doing my research I found that there were a lot of things that when you overlaid a different graph you might not have ever really expected that it did some bumps. Ilian was talking to me about how he looked at Block N Load’s CCU, but he overlaid when a popular YouTube video was uploaded about it. And you can see it would definitely spike up exactly after that video was posted.

Erik: That’s the difference between localised effect and the entire lifetime of the product. Of course, that’s what you’ve shown in your thing. Obviously, a new update will create an instant CCU but whether that CCU will be additive or just a bump, is going to be left to the quality. We’ve got Block N Load here, so you see that. This is what Block N Load looks like and you may or may not know that we're trying to actually do something about Block N Load lately. Now, the big update we did just virtually had no positive effect whatsoever.

You see that but in the grand scheme of things, the work we've done in the last six months has had very little impact. And we've done, what, three, four releases in the space of two months, with monetisation and everything, that actually have had no effect whatsoever on the game.

Josh: And do you know why?

Erik: Yes, partly. But we never know 100% why but partly it's the fact that the OGMs for Block N Load, which we pulled probably a little bit too late in the process, is essentially children.

Josh: Is what, sorry?

Erik: Children. And I mean by children, less than 15 years old; 50% of the audience of Block N Load is less 15 years old and 75% less than 18 years old. So we're talking about typically more difficult to monetise audience, especially when you're talking about a free-to-play game where you basically wouldn't imagine a child asking their parents, because they wouldn't be able to pay for themselves, a repeated amount of money to open loot boxes or those kinds of things.

It's going to be very different is you're talking about Minecraft, which has basically the same profile and appeal to a younger audience but it's a paid-for product. Therefore, you can have a child saying to their parents, "I need £20 to buy Minecraft." And it makes sense for a parent.

But Block N Load, if you try to actually monetise a free-to-play environment and if you imagine the child asking their parents, "I need £20 to buy a loot box." It won't work and part of it is the age. And second part of it is the providence. It's become very popular Russia and in Eastern countries, which are notoriously less monetisable than Western countries, like the US and the UK.

Josh: When I've been looking at the data I've done it with the assumption that you would like to maximise CCU, you'd like to maximise the positive sentiment and then chiefly you'd like to

	maximise the money made off that. From what you're saying is that sometimes one doesn't necessarily feed into the other. If you had a really high CCU still and the audience was still too young, it still might not translate into log pin sales.
Erik:	Well, I don't want to give too much detail about that. I'm trying to hide that, but if you see this kind graph that I'm working on - I open that. This is a complex model where basically you start with the number of players you have and then you figure out how many speak about the game to their friends. And you see all these curves that I showed you feed that, and try to say this is a typical evolution of how many daily active users. And then you figure out how many of these daily active users are converting into concurrent peak users or average users, and you've got this equivalent curve. And by 30% of this activity you get your CCU. But since every green box you see here can have a different evolution, it can be any of the things that I got then everything is the air.
Josh:	So typically you've already got all of the information. You can't really make any-
Erik:	It's less that rather than the extreme volatility and chaos theory of it. It's the butterfly effect, really. I change one percentage in that sheet and the game becomes suddenly fantastic. And that's the problem we have forecasting and the kind of analysis that says, a benchmark says these standards. But the benchmark says that's what Block N Load does. Should we take that as a model? It's a peak and then very, very stable, 2000 CCU. That's not very exciting. Should we take LawBreakers as a model? Look at that. Should we take Warframe as a model? That's more interesting but it requires basically dedication and it's five years of Warframe. Then the question comes back, how credible is this information compared to, do you have the feeling it's going to work. And I'm talking to you after several years of asking myself the same question. Several years of trying to figure out the holy grail of how can we predict the revenue of a game and I just come down to the fact that probably, the best way is to actually feed on the evolution of others and figure out maybe it's an average of all

this. But give me the average of that per game and you've got something that's actually making no sense.

Josh: There are a lot of variables.

Erik: That's what I'm talking about. So the amount of variables all then govern halfway through the marketing and the talent of the team, but also external trends, competition and pure luck; exposition, timing. You're looking at the future. This is one of a thousand possible futures for whatever game you're looking at.

Josh: That rings true with some of the stuff that I found as well. There was a master's thesis I read about trying to predict a game's success, but it wasn't really useful because the accuracy was questionable. Or also, the outcome was trying to predict the metacritic score. But that doesn't necessarily-

Erik: I know. It's exactly what I'm talking about. And remember, this is-

Josh: Deep Gabe I think the guy called it.

Erik: Games Foundry, which basically I've met some time ago. And the interesting thing is - no, it's not this one, something Foundry. Digital Foundry? No it's not, whatever. Basically, there's a system that tries to actually list all of the features that you have in your game and try to predict how successful it's likely to be because it has this feature and this feature and this feature. It's ludicrous. And they tried to convince me that yes, machine learning. It's statistic and so on. It's command.

Let's just step back and figure out that actually, it's just not because you have these features it will make a game great, but that's some people thinking. If you come back in terms of NPD and that's what we've tried to do with some of our projects, is to try and say there's no recipe. It's just like in music, just like in art, there's no recipe. If you just put it out there and figure out if it resonates with people. That's what early access is about.

The earlier you can actually validate that your idea or concept that has an appeal has got a resonance with your target audience the better you are because you can build on this. If you try to release something that actually has no appeal you will know very fast and will have wasted less money than if you spent one year developing the game of your dream and then realise no one wants to play it in the first place.

Josh: When I was doing this, I think I started out grand as well. I started out really broad. I was going to predict what makes a successful game and then immediately was shot down, and rightly so. I've ended up actually looking less at about trying to predict what makes a game successful but rather very, very focused answers to some very focused questions might be in regards to the actual development process of a game. In any game or any software product during that period of metamorphosis there are a series of decisions you can make. And just trying to give a little bit of feedback as to what you might do based on what other games have done or what other products have done, so update. How often to update and how big it should be?

Erik: Yes, just going back again, trying to compare apples and oranges all the same. One of the games that I like is, apart from Warframe, which is almost too good to be true, because it's been quite stable for several years. It doesn't give us much information, but it's Rainbow Six Siege. That's really what is interesting, is the fact that for a year nothing happened. It was basically a game worth nothing. It was crap. It was buggy. It was not working. Then they kept working this.

It's not like they needed to update it. The updating cycle was bringing the game to the point where it is today. And you see it's not quadruple-

Josh: It's building a better product, that it makes sense that-

Erik: Yes, and the problem is, it may not work ever again because the pure, I would say that the driven kind of approach, the more you update you see an update has a positive impact on things. First, I proved with Lock N Load it's not always true. Second, it all depends on the quality of the update-

Josh: Or even just what it is, the content.

Erik: What it is, exactly.

Josh: I looked at instances where a really massive update didn't really have any effect because the players didn't care. Like, Menaphos got lots of play and they get feedback, and it was a massive piece of content. But then, one thing that fixes a UI bug that was small but people really cared about it suddenly gets massive in play.

Erik: You know what, all of it, when we come back to the root of it, is what player obsess is. Player obsess is not functioning, as we should say, for Jagex, is every single company in the world should be player obsessed or consumer obsessed. The reason for a company to exist or for any artist who want to be paid for their work is to figure out what do you want. Can I create something that you do really want, what evades you and actually have something there for you to pay me enough so that I can make a living out of it. So a company is the same. It's basically marketing 101. It is understanding your player and what they want. What is exactly the next thing they would love to play and are you able to engage them. And this doesn't obey curves. It's human psyche.

Josh: That feeds really well into my last question, which you've started to answer already. But I was going to ask, is the development of new content, once you've already selected an idea and you're running with it, and you're giving it an amount of time to grow and perhaps become successful. Would you say that the development is community sourced? How does the community feed into what you do?

I know we said things like RuneScape are obviously almost entirely based on the community, but with NPD specifically, is it?

Erik: It's exactly that. The more we can the better. It's not exactly easy because there are a number of things - there's also old thinking about you want to keep things under cover because you don't want competitors to jump on it and steal it from you. That's the typical thing. You may want to release a great game idea and people will figure out, okay, that's successful but we are much more powerful, much more nimble and we'll develop a better product faster. It's almost what happened to Battle Royale and Fortnite. Basically, PUBG wanted to sue Fortnite for making the Battle Royale Game.

Josh: The pot calling the kettle black.

Erik: Yes, ridiculous when you think of it but that's really what it is. So some people might want to say, "Hey, we don't want to actually share our ideas with the world yet because someone would be able to steal it from us."

Josh: So would that prevent you from giving early access as a public platform?

Erik: To a certain degree, yes, and to another, best find the balance. Because if the idea is scrap, no one will copy it anyway and if the idea is great you already have got a length of development and thinking about it.

Josh: But I suppose even just alphas in some way constitute early access, like giving it to a selection of-

Erik: The interesting thing is, ever since I've been in this industry and even before when I was working as a consultant in business-led companies developing software, not games. God knows what an alpha is. God knows what a beta is. God knows what early access is. God knows what a full release is. Basically, can you use the product or not. And for that matter, it's always been interesting to see that, again, a lot of developers say, "Well, on a console they don't really understand pre-access. They don't understand early access. They don't understand the game is not full and ready to use." As long they can play they will say, "It's buggy." And the fact that you called it beta or early access wouldn't change a thing because it's buggy.
So there's the mentality of basically people saying, "It's not ready to release," and people would complain about issues that are not supposed to be ironed out. But that's the point. For me, you draw a line in the sand. Regardless of how many, how confident it is, people start playing it and in that second stages how do you restrict the access is the key point. If it's still a plug, if it's still basically a place where you have to actually show you want to sign a NDA to be able to play, you are still in a controlled environment.

As soon as you make it open, open alpha, open beta, you name it open or early access, whatever, then it's in there. You can call it early access, people play it and DayZ has been in early access, or Fortnite is still in early access as we know it.

Josh: And it's those people who are playing the game that then not solely drives the rest of development but certainly has a massive impact?

Erik: Yes, exactly. I want to say it's between solely and massive impact because if it's solely, of course it's almost the same thing. Maybe the people who play the game don't know what they want. They can't express it. They are not exactly sure what they want and even if they want something, it may not be good for the rest of the people. So you just have to actually sift and select what design feedback, some concept actually makes sense within the vision of the game as long as the vision of the game is capturing the necessary audience aspirations or motivations.

Josh: So the data that I've got, although we've talked a bit about apples to oranges, how one product is vastly different from another and has very different variables. Certainly the context for one early access game and another early access game in the scope of the platform released on Steam has the same store page format and things like that. Would you say that there are still some valid comparisons or things to look at between them, like, they all have a bank of reviews? They all have updates that are applied to them? That they share context and they share the publication and the platform, but they don't share the artistic ideals or they don't share the community?

Erik: Yes, I agree in many ways. It all depends on what you want to look for in your data. Especially don't try to actually make your data say what you expect to say. Just look at your data independently and say, "I see a relationship between this and this." Maybe test it, figure out if it's actually true or if it's not. As a data scientist you really want to say, "I have this hypothesis. I will either contradict it or approve it, but I don't have a preconceived notion of what it should be."

Josh: Do you think it will always be bespoke? Do you think it will always be dependent on the sort of game, like there's no detaching it from its content, as it were?

Erik: Yes, I believe so.

Josh: I agree as well.

Erik: Let's step back. We come back to the fact that as an entertainment medium, video games are just like movies. We are just like music. Now, you can't go to a musician and tell them, "You need to make it work like that. You need to go from C to A and then you come back to B Minor. And that will create a hit."

Josh: I can only think of one instance where an artist has released an album early and then developed the songs, and worked on it.

Erik: Who was it?

Josh: Kanye West released an album called The Life of Pablo. He released it at some point and then would tweet every day, "I'm going to listen to what you said about this track, and I'm going to go back and change the track." And because he was publishing it almost, I think it was purely digital, he was able to just replace the track. And interestingly, it actually resulted in, like, almost strangely similar, people have tended towards positive reception at the time.

Erik: That's exactly that. The difference between music and games is the fact that, how can I put it? Music is usually more the expression of an artist and games is more an entertainment product. You have the same thing for movies in the midway. You basically have pre-screening for movies that actually come back. Typically Star Wars is really known for that where they project it to a specific audience and figure out the specific audience didn't get that or didn't like this, so they redid the movie in a different way so that actually it hits better scores. It happens everywhere in entertainment but to a degree only.

Josh: Would you personally remove from this set of information the things that aren't games them? If I was looking specifically at games, because I've got about 70 non-games, like audio software, video production software that are now being published through Steam, would you say they're too different then as a software product versus this thing that is inherently

	artistic and creative? Could the same comparisons or insights that were drawn from a piece of software be applied to-?
Erik:	That’s an interesting one.
Josh:	There might be an area of study.
Erik:	Possibly. The reason is what I said. As soon as you’ve got art you’ve got an expression of sensitivity or that sensibility that actually, it makes people like it or dislike it but on the software it’s much less so. Some software does that pretty well and I’m looking at Slack, for instance, so software that actually has got personality to some degree. I’m thinking about Tinder or things that basically have got a feature that makes them different, Snapchat. You think of these kinds of software they’re not games. They’re applications that have a personality to them, Spotify, you name it, where basically, if you’ve got a certain artistic intent. Now, let’s come back to the Apple products, for instance, where design-driven Apple products have got something to say. They’ve got an artistic statement, as well as a service to the customer. Ajeera would be much more on the software side to say, “Well, we’re not trying to be fancy. We are not trying to do something that you feel exciting.” I look at Ajeera and what I see here, if I go back there, is very different from Trello. I’m not sure if you know Trello?
Josh:	Yes.
Erik:	And Trello is that. And you see the difference in how it translates into the psyche of a customer or the player, the customer in terms of how engaging the product’s become.
Josh:	But I suppose from the context of publication Jira and Trello will have been published at one point, and somebody in the company will have had the intention to add lots of features in the same way that a new game is developed and has the intention to add more mechanics.
Erik:	Yes.

Josh:	And over time, as things are added, hopefully it improves and then tends towards - and that feedback from the person using it might be similar. I go back to Minecraft after two years and I still go back because I'm curious as to what's been added, and it's made it a better product. And I might go back to Trello and go, "Oh my God, they've added G Drive! That's useful to me and I can use them more." But yes, this has been fantastically useful.
Erik:	I hope.
Josh:	Thank you so much. I'll stop it now so that you're not under the gun.

9.3 Industry Survey Results

Below is a full account of the survey with Joel Graham used to evaluate the project.

Disclaimer

The survey will be conducted in one session with both parties present in order to ensure that everything is understood and recorded. I will ensure that my paper lists as one of its aims to evaluate the findings in an industry context.

Intention

This intention of this survey is to evaluate all critical aspects of the dissertation project including its justification from a business perspective, the methodology by which it was undertaken, and finally the applicability of its findings. This will furthermore be used to help assess to what degree I achieved my functional and personal goals.

Meta

The participant will be provided access to the database, lab notebooks, a summary of the results and the project poster.

Survey

Name: Joel Graham
Role: Director of Analytics and Data Science

S1 Justification

The intention of the project was to improve the new product development lifecycle for software projects that will receive an early release by examining analytical data from a large cohort of similar existing products. *Describe research as needed.*

S1.1 How would you usually aid a project team looking to develop and release a new project?

> Development is expensive, and returns in the games industry are wildly variable. There is significant benefit in being able to validate early, and use data from that early release to give a better idea/projection of likely longer-term success = is it worth funding.
>
> Games are a creative industry though, and robust data to link early release to longer-term success is incredibly hard to uncover: with the advent of Steam and their Early Access programme though, we have a sense of their being a great opportunity to for the first time have a large dataset that could provide answers to some of these questions.

In the past, we have been limited to looking at the trends in CCU or owners over time, for our own products only, without consideration of the wider market and audiences - and we recognise the limitations of this, particularly as the business evolves, e.g. changes in genre of game released.

S1.2 Do you see a need for research in this area (NPD)? Would you plan for that or undertake it ad-hoc?

Yes | No

Yes. See above.

I would anticipate that this could only provide real insight if given a dedicated and substantial block of time devoted to it - the benefit would come from the review being a comprehensive one, so I'd expect to need a commensurately large amount of resource to achieve this.

S1.3 How much information do you share with other companies in regards to the successes/failures of this process?

Everything | Significant Amount | **Some** | None

S1.4 There are many issues present in software NPD, do you think that this kind of research could help in some way?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

S1.5 Could you see a use for such research at your company?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

S2 Methodology

The data has been collected from the platform 'Steam' products tagged as 'early access', and pre-processed and funnelled into the document-database 'Couchbase' using a microservice architecture ('Python') cloud-deployed using 'Docker'. The investigations were hosted using a Jupyter notebook kernel. Key quality aims here were scalability, reproducibility and transparency. *Describe technology as needed.*

S2.1 Is the dataset relevant to you? (is it cross-applicable, or perhaps too rooted in the context of the system?)

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

S2.2 Do you think the dataset has sufficient breadth (representation)?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

S2.3 Do you think the software choices were appropriate? How do they differ from your team's?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

The choices were definitely appropriate - they differ from our team's only slightly, and in the main because we know with our internal work that there is a fixed and internal-only group of collaborators all of whom will have access to the same network (as well as needing to maintain certain security concerns). Given the open and collaborative nature of this research, which is itself a benefit to the research, the choices were right.

S2.4 Given access to the project repository and subsequent tooling guide, how confident are you that you would be able to utilise the work of this project to your own benefit (whether that be reproducing the same dataset to perform your own analysis, or adapt it to operate on new data). Are there any organisational issues that would hinder this?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | **Agree** | Strongly Agree

The setup, source data, technologies and methodology are a perfect fit for how we work and the way that we document research for potential re-use by others. The only key addition that I'd typically look for would be structure and recorded information with regards to data munging/ETL that took place between the original source data collection and the final groomed / cleaned reporting data set.

S3 Findings

The analysis was undertaken iteratively. First, the features were examined for bias and to prove/disprove assumptions. Next, they were used to examine the factors influencing user sentiment over time. Next, I attempted to quantify the impact of product updates on these features. Logging the results and what they relied on at each step, I have produced a set of factoids to help direct the course of the NPD lifecycle.

S3.1 Do my findings align with your own expectations?

Strongly Disagree | Disagree | Slightly Disagree | **Slightly Agree** | Agree | Strongly Agree

The findings are in many cases a validation of my pre-existing expectations - which is valuable both as a reassurance that our existing and less formal / objective domain knowledge that we have been using to make decisions is accurate, as well was providing a methodology to maintain an objective view on the continued truth of this (e.g. if the market changes over time, this research can be regularly re-run to update those expectations). In addition, and even more valuably, some of the findings are substantially different to my pre-existing expectations. In those cases, I could see us using the findings to make fundamentally different decisions about the path of product development/release based on early metrics - which is a great goal to have accomplished to any level!

S3.2 Could the findings help inform some aspect of strategy at (or a hypothetical) company? Would you feel confident using this research to contribute towards informing decisions?

Strongly Disagree | Disagree | Slightly Disagree | Slightly Agree | Agree | **Strongly Agree**

S3.3 What do you see as the most useful and least useful things?

Most = bias of scores among players who received products for free; the impact of updates upon scores and engagement; abandonment vs. completion is also an interesting addition. Least = some of the general quantifiers of e.g. positivity overall / trend over time (as those are already discernable from public resources and simpler analyses that have been shared historically)

S3.4 Do you see further investigation as worthwhile? What is missing, or needs elaboration?

Yes. An ML approach to feature importance ranking would be great; further attention / multiple second opinions on the ETL process would help reassure as to the robustness and replicability of it; and for long-term re-use, tracking the evolution of the market/audience over time (which to be fair could just be done implicitly by the use of calendar date/epoch as a feature) would help understand how the features change when e.g. market saturation or Steam storefront changes take place.