On the Impact of Software Patching on Gameplay for the League of Legends Computer Game

Mark Claypool, Artian Kica, Andrew La Manna,
Lindsay O'Donnell, and Tom Paolillo
Computer Science Department, Worcester Polytechnic Institute
Worcester, MA, USA
email: claypool@cs.wpi.edu

Abstract

Traditionally, most software patches have primarily fixed bugs or improved performance in previously released software. However, modern computer games use patches not only to fix software but to change gameplay, as well. Despite the importance of changes in gameplay to both the players and the game designers, there is little or no published results nor 3rd party Websites that analyze game data along with patch data. We analyzed the effect of patches on gameplay for what may be the most popular computer game in the world, Riot Games *League of Legends* (LoL). Our methods: 1) harvested all available patch data – over 160 patch files with over 7700 patch changes – and created a novel taxonomy to classifying the patch changes; 2) gathered game statistics from over 11,000 players and over 465,000 games; 3) analyzed game data and patch data separately and in combination to gather insights into the impact of patching; and 4) developed a publicly accessible Web site that allows for interactive exploration of game data and the corresponding patches. Analysis of the data shows that Riot Games applies patch changes to LoL gameplay twice each day on average, about 10x more often than Riot Games patches LoL bugs. Patches tend to keep player-chosen champions close to a win rate of 50%. While most gameplay patch changes can be categorized and quantified, the effects of patches are not straightforward and require interpretation of patch text in order to fully understand their impact.

Keywords– game, analytics, co-op, game balance, patch

1 Introduction

Usually, software is released with all major content and features in place and, hopefully, most bugs removed. If fixes or improvements were needed, well-known software engineering techniques to fix (or "patch") were used in which the software and/or its supporting data was updated. However, while traditional software patching was used primarily to fix bugs or improve performance, many modern computer games use software patching to adjust game balance and add content to the game. In fact, many games are intentionally released without all game content in place and even without all gameplay elements fully vetted. Game patches are subsequently pushed out with additional game content, adjusting games rules and game balance which effectively alters the gameplay for current players.

A popular example of just such a game is *League of Legends* (LoL, Riot Games, 2009). LoL is perhaps the most popular computer game in the world based on hours played per month (Raptr - The Offical Blog, 2015), with 27 million players each day and 7.5 million players simultaneously at peak times (Riot Games,). As part of the eSports scene, LoL is lucrative for players, too with a professional player circuit and tournaments with prize pools over $2 million (E-Sports Earnings, b; E-Sports Earnings, c), among the largest in competitive gaming (E-Sports Earnings, a).

Since its release in 2009, Riot Games has patched LoL over 160 times,[1] an average of about 1.5 patch files per month. LoL uses both traditional game patches (that both fix and improve software) as well as modern game patches (that release new game content and adjust game balance). These patches have introduced significant amounts of new game content, roughly doubling the number of avatars (called *champions* in LoL) available to players since release, and altering the game balance for active players. Adding game content and changing game balance this way is critically important to both players for making the game fun (Claypool et al., 2015) and for game designers for keeping the game engaging. Despite this importance, there has been little formal analysis of the effects of patching on gameplay.

Riot Games provides access to extensive data on previously played LoL games through an online Application Programmer Interface (API).[2] There are numerous third-party Websites that use the Riot Games API to present game data that is helpful to both players and game designers. For example, Champion.gg (cha,) provides data on in-game choices players make and their impact on winning/losing games and LolKing.net (lol,) provides some of the same data as well as a summary of the latest patch notes for each champion. While many users may only wish to know about the latest (patched) game version, studying past patch changes can help players and game designers predict how current patch changes might affect the game.

Unfortunately, to the best of our knowledge, there are no third-party Websites that have historical game data going back more than a few months nor are there third-party sites that analyze patch content in any quantifiable way. Lacking from all third-party LoL Websites, and published literature for all games, are correlations of game data with patch changes in order to better understand the implications that patches have on the game. As-is, third-party Websites require users to manually correlate patch data, if patch data is even provided, with game data and make it impossible to do so for historic game data.

In order to analyze the effects of patching on League of Legends gameplay, we first developed a method to gather a large, random sample of game data from the Riot Games database. Our gathering process yielded data on over 11,000 players in over 450,000 ranked games from Seasons 4–6. We also developed a method to harvest and classify LoL patches, devising a novel taxonomy of the types of changes relevant to fixes and gameplay changes. Using our method, we extracted and classified all available LoL patches from pre-release Alpha through Season 6, yielding over 7000 patch changes from 164 patch files. In order to

[1] `http://leagueoflegends.wikia.com/wiki/Patch`

[2] `https://developer.riotgames.com/`

allow for interactive exploration of the patch data in relation to the game data, we developed a novel Web site[3] that enables users to select LoL champions and study how patch changes have impacted their performance over time.

Our analysis of game data shows the win rates for champions chosen by players to be mostly normally distributed around a mean of 50%, but with fewer champions much higher or much lower than this rate, probably due to patching. Champion pick rates are not at uniform, with a heavy skew to the most popular champions. Ban rates are even more skewed, with a small percentage of champions being banned most often.

Our analysis of patch data shows a fairly steady rate of changes which would not be expected for a mature, stable software. The majority of changes are not fixes or visual improvements, but are changes to the gameplay itself in the form of adjusting balance or adding new content.

Correlating game data with patch data shows changes, indeed, tend to increase win rates for champions with lower than average win rates while reducing the win rates for champions with higher than average win rates. However, more detailed analysis also shows that simply counting changes does not always accurately predict the effects on win rates.

Our work makes several key contributions, providing: 1) an aggregate analysis of the most important in-game statistics for LoL players, accompanied by a Web site that enables interactive exploration. While other third-party Web sites have some of the same content, we are the first to provide such information in a historic context for comparison; 2) a comprehensive classification of all LoL patch notes, categorized into a novel taxonomy that allows for exploring the relationship between the kind of patch and the effects on gameplay; and 3) an analysis of the correlations between patch data and gameplay.

The rest of this paper is organized as follows: Section 2 describes background and related work; Section 3 details our methodology towards gathering patch and game data for LoL; Section 4 analyzes the gathered data; Section 4 presents the LoL Crawler Website for interactive exploration of LoL patches and game data; and Section 6 summarizes our conclusions and provides for some possible future work.

2 Background and Related Work

This section first provides background information on League of Legends (Section 2.1) (LoL) and third-party Web sites that provide data on LoL (Section 2.2). The section next presents patching as is relates to LoL (Section 2.3) followed by some examples of research work that analyze software patches in other contexts (Section 2.4).

2.1 League of Legends

League of Legends (LoL) is a multi-player online battle arena (MOBA) game where players are matched, based on skill, into 2 opposing teams. While there are several game variants for casual players, most competitive (also called *ranked*) matches and all professional matches are played on a standard game map with 5 players on each side. The objective is to destroy the opposing team's headquarters. Once matched to a team, each player chooses one out of

[3] `http://lolcrawler.cs.wpi.edu/`

128[4] different characters, called *champions*, to control during the game. Gameplay consists of controlling the champion to fight opponent champions as well as other lesser monsters on the game map. Champions vary in their in-game abilities (e.g., method of attack, types of spells) and attributes (e.g., amount of damage, rate of attack) and are enhanced by skills and items gained during the game.

In ranked games, players take turns picking champions and there can only be one of each type of champion in a game – i.e., the end result is 10 unique champions each game. Before champion selection, each teams bans 3 champions which prohibits the other team from choosing them for the game – i.e., there are 6 unique champions banned each game.

The best LoL teams are those whose players cooperate well, with players using their champions to fill roles (e.g., a tank to absorb damage or a healer to aid wounded allies). In fact, Riot Games designs champions with specific roles in mind. The most basic roles include damage dealers, tanks (champions that can absorb a lot of damage), and utility (champions that can heal teammate champions or that can control enemy champions). Some champions can fulfill different roles depending on how they are played and the in-game items purchased.

The specific champion roles Riot Games identifies (Picture of Horse, 2016) include:

- *Assassin* – can defeat an enemy champion quickly with combinations of attacks and abilities.
- *Fighter* – attacks at close range, both absorbing and dealing out damage.
- *Mage* – uses combinations of abilities to maximize damage.
- *Marksman* – attacks from distance with consistent, high damage.
- *Support* – assists other champions through abilities (e.g., healing).
- *Tank* – absorbs damage with defensive attributes and initiates combat.

League of Legends only allows players to compete against other players in the same geographic region (i.e., connected to the same server). Regions include Brazil, Europe East and West, Japan, Latin America North and South, North America, Oceania, Russia, and Turkey. Most competitive players are ranked into tiers of Bronze, Silver, Gold, Platinum and Diamond with 5 divisions in each tier. The vast majority of ranked players are in the Bronze through Platinum tiers. For the superlative players, there are two elite tiers, Master and Challenger, that have one division each, and the Challenger tier has the best 200 players on that server.

2.2 League of Legends Analysis

There are numerous third-party sites that provide data on League of Legends gameplay and champions. This section introduces two popular sites by way of example that illustrate their features as well as their shortcomings in addressing our research problem.

[4] As of February 2016, the time of this study

Figure 1: Champion.gg – a third-party Website that provides statistics on champions, as well as other game data.

Champion.gg (cha,) is a third-party Website that presents data about each champion, including current win rate, play rate, and ban rate (i.e., the percentages of games in which that champion wins, is picked and plays, or is banned). A Champion.gg Web page is depicted in Figure 1, showing champion win rates. The column on the left has champions in each of the roles (e.g., Support champion Sion with a win rate of 55.2%), and the column on the right has the same breakdown, but for the lowest win rate (e.g., Support champion Nunu with a win rate of 36.9%). Champion.gg also provides basic performance statistics such as average kills, deaths, and assists earned per match, with comparisons to other champions. While Champion.gg does provide some historic context, data is only provided for the most recent 5 patches (about 2.5 months). In addition, Champion.gg does not show the actual changes, known as patch notes, made to the game which requires players to do any sort of patch analysis of patch changes and, for example, win rate data, manually.

LolKing.net (lol,) is a third-party Website that provides a searchable database of players and champions, and acts as a hub for discussion and activities regarding League of Legends. Relevant to this section, LoLKing shows the patch information associated with changes to champions. An example for Patch 6.20 is shown in Figure 2 where changes to champions Ashe and Cho'Gath are summarized in the text right under their names (e.g., for Ashe, "Crit slows are stronger but decay faster. Q is no longer a full attack reset"), and details are provided at the bottom of their profile (e.g., "Frost Shot (passive) base slow scaling tweaked from 5-25% ..."). While helpful for players to better understand how a champion may have changed in the last patch, lacking is any way to link the patch changes to statistics (e.g., win rate) on the champion.

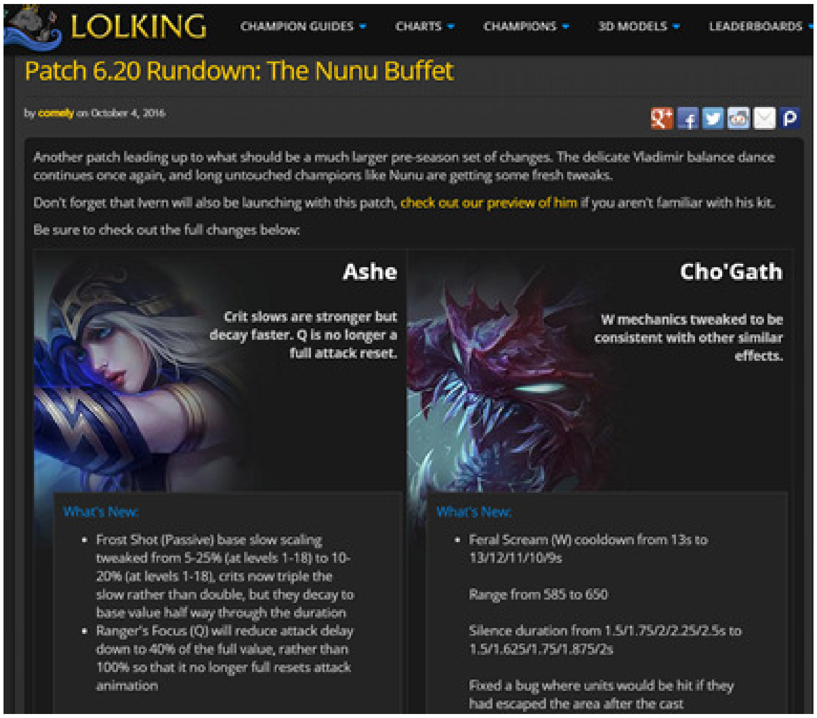

Figure 2: LoLKing – a third-party Website that provides path notes for each champion, as well as other game data.

2.3 Patching

In games, "patching" is the process of changing an existing game. While there are thousands of changes made to a game before it is released, after release the changes are added on top of the existing game from the player's perspective, hence the name.

Many modern day games develop internally until a playable version of the game can be played externally, even if the game does not necessarily have all planned content. The developers deliberately intend to patch the game with both fixes to existing problems as well as add additional content as the player base grows.

For League of Legends, patches contain (pat,):

- New game content – new or remade champions, items the champions use in game, and skins (visual changes to a champion).
- Balance changes – changes made to even out the power balance among champions.
- Bug fixes – changes made to correct unintended behavior in the game.
- Technology improvements – anything new for better performance in the game lobby or in the game.

While bug fixes and technology improvements are likely welcome by nearly all players, new game content and balance changes are different. Effectively, new game content in the form of new champions, new items, new abilities and even a new game map, require a player to re-learn aspects of the game that s/he may have already mastered. Balance changes are typically somewhat easier to re-learn since the gameplay remains the same, just the relative effectiveness of different actions changes, but they still require adjustments.

As an example illustrating League of Legends patch content, consider a patch for the champion Katarina[5] in patch v3.01.[6] As with most significant patches, here Riot Games provides a narrative with rationale for the change to a champion followed by details of the patch:

> "These changes are focused on once again increasing the risk of Katarina's gameplay by forcing her to build more offensive stats and focus on pulling off well-timed Death Lotuses."

Bouncing Blades initial damage reduced to 60/85/110/135/160 from 60/90/120/150/180.
Shunpo damage reduction duration reduced to 1.5 seconds from 3.
Death Lotus total ability power ratio increased to 2.0 from 1.75.

The first two changes decrease the strength of the champion by reducing the damage of the Bouncing Blades attack and damage duration of the Shunpo ability, while the third change increases the scaling power of the Death Lotus ability. As the narrative indicates, the changes are not only intended to merely change the relative strength of Katarina (i.e., not just making her stronger or weaker), but also to change the way she is *played*.

Thus, it is clear that Riot Games patches are not only intended to fix the game, nor even to just balance the game in terms of champion strength, but are also meant to change the way the game is played. This is apparent in a quote from a Riot Games developer:

> "One of our biggest philosophies for League of Legends is we want to keep the game fresh, we want to keep the game changing..." – Patch Rundown v5.16[7]

2.4 Patching Analysis

There has been research on traditional software patches e.g., (Schryen, 2009), but not as they relate to games and not with any analysis of the effects of the patches on players/users. Systems-type research (Gkantsidis et al., 2006) has used data traces to analyze the effectiveness of different strategies to deliver software patches, including client-caching and peer-to-peer, but does not analyze the impact of the patches.

While there is considerable research on the use of patches to address software security, e.g., (Arora et al., 2010), such patches do not typically enhance the patched software with new features. Other related work has designed a system that automatically generates patches for specific bugs that make the software prone to security breaches (Lin et al., 2007), and has evaluated how many real-world systems may be vulnerable to such exploits.

[5] http://gameinfo.na.leagueoflegends.com/en/game-info/champions/katarina/
[6] http://leagueoflegends.wikia.com/wiki/V3.01
[7] https://www.youtube.com/watch?v=DH0HpI-PqJ0

For computer games, security patches are often applied to fix code so that known “cheats” (exploits that players uncover to gain an unfair advantage) are no longer possible (Webb and Soh, 2007).

There is some published research on data analytics, e.g., (Hullett et al., 2011), an important area for game development, but none of it has focused on analyzing patches for games nor downloadable game content.

The prevalence of downloadable game content (also known as *DLC*) is well known to many players (Lizardi, 2012), but to the best of our knowledge, there has been no systematic classification and categorization of such content nor relation to gameplay. Similar to downloadable content, modifying an existing game (or just *modding*), refers to custom alterations to existing games (Nieborg, 2005). Such modifications are often applied as a patch – in this case, a patch that is produced by other players rather than the game developers. Research in modding has examined the diversity of mod culture (Kow and Nardi, 2010), identified potential future directions modding may take (Sotamaa, 2004), and described how publishers can profit from creativity that arises from mods (Jeppesen, 2004).

3 Methodology

Our methodology to analyze the effects of patching on League of Legends (LoL) gameplay is as follows:

1. Collect data on in-game LoL attributes (see Section 3.1).
2. Harvest data on LoL patches (see Section 3.2).
3. Build a Web site to allow for interactive exploratory analysis of game data and patch data (see Section 4).
4. Analyze both game data, patch data and combinations (see Section 5).

3.1 Gathering Game Data

Riot Games provides access to game data through their Application Programming Interface (API),[8] a method to access structured game data in a secure and reliable way. Once a developer key is obtained (free with any LoL user account), access is gained through HTTP GET requests. For example the command:

```
curl -X GET https://s3-us-west-1.amazonaws.com/riot-api/seed_data/matches1.json
```

uses a command-line tool to request data from the indicated URL. The output is returned as plain text in JavaScript Object Notation (JSON):


```
{"matches":[
  {"matchId":2054994244,"region":"NA",
   "matchDuration":3067,"queueType":"RANKED_SOLO_5x5",
```


[8]https://developer.riotgames.com/

```
    "season":"PRESEASON2016",...
  ...
}
```

The example shows the start of a series of games (matches), each with a unique id. Games have statistics such as region (e.g., "NA" for North America), duration (e.g., "3067" seconds), game type (e.g., "RANKED_SOLO_5x5" meaning a competitive) seasons (e.g., "PRESEASON2016") and other attributes not shown here.

The API provides complete data on ranked games for each player from Season 4 onwards.[9] and includes information on champions picked and banned as well as which champions won and lost. Some of our future work may include gathering and analyzing additional information available through the full API reference,[10] such as scores, damage dealt and received, and items purchased. Game data is available for normal (non-competitive, non-ranked) games, too, but only the most recent 10 games for each player is available.

Since the default API key only allows a maximum of 10 requests every 10 seconds and 500 requests every 10 minutes, Riot Games approved our request for a special production key that enables 3000 requests every 10 seconds and 180,000 requests every 10 minutes.

The API provides no built-in functionality for extracting a random game nor even a game with specific attributes (e.g., player ranking). Nor is retrieving the full population of games feasible, even with our production key, given that there are over 1.5 million players on the North American server alone, many with hundreds and even thousands of matches in their histories. Instead, a novel method to gather a representative sample from the population of all ranked games is needed.

Our approach to sampling proceeds with "seed" players – 25 players, one from each division from each tier[11] – randomly selected from players listed at LolKing,[12] a third party Web site that lists all players by tier and division (see Section 2.2. The LoL API provides the history of ranked matches for each seed player, which is used to collect additional id's of players that competed with the seed player, hence are likely have similar player rankings. The process is repeated until there are enough unique players selected. From this full set of players, game histories are combined, duplicate ids removed, and a random sampling of games chosen from this combined pool.

Once the list of games was compiled, the full match details were pulled from the Riot Games API, and game statistics computed from the extracted data. In our case, we computed data on champions – winners, losers (both of which were "picked") and those banned.

3.2 Gathering Patch Data

Riot Games provides patches that change many aspects of LoL all at once, including changes to champions. We use the term "patch change" to mean a single, individual change to a champion. For example, a single patch change may increase the health of a specific champion. We use the term "patch file" to mean the group of patch changes released by Riot Games. Thus, a single patch file typically contains many individual patch changes.

[9] Prior to Season 4, game data is incomplete and win rates for champions cannot be computed.

[10] https://developer.riotgames.com/api/methods

[11] Excluding Master and Challenger, since the vast majority of players are in Bronze through Platinum.

[12] http://www.lolking.net/

Table 1: Summary of League of Legends Patch File History

Season	Version	Patches
Season 6	v6.x	1
	Pre	v5 22-24
Season 5	v5.x	1-21
	Pre	v4 20-21
Season 4	v4.x	1-19
	Pre	v3 14-15
Season 3	v3.x	1-10, 10a, 11-13
	Pre	1.0.0.152 1.0.0.153 1.0.0.154
Season 2	v1.0.0.x	126-139, 140b, 140-146, 146b, 147-151
Season 1	v1.0.0.x	32, 52, 58, 61, 63, 70, 72, 74-75, 79, 81-83, 85-87, 94b, 94, 96-116, 118b, 118-125
Closed Beta	v0.9.x	22: 4, 7, 9, 15, 16, 18, 21, 24, 34
	v0.8.x	21.110, 22.115
	2009-x	04: 11, 18, 25; 05: 1, 9, 15, 23, 29; 06: 6, 12, 19, 26; 07: 10
Alpha Stage		Week 2-7

Table 1 summarizes the LoL patch file history, aligning the season with the patch file numbers. As seen in the table, since LoL's initial release, season and patch file numbering has changed. To be consistent, our work refers to seasons as Alpha, Beta and Seasons 1 – 6 and numbers patch files consecutively from patch file #1 at Alpha Week 2 to patch file #164 at Season 6 Patch 1 (v6.1).

Unlike LoL game data obtained through the Riot Games API, patch changes are not provided in any structured form. Instead, patch changes are released only in human-readable text. Fortunately, patch changes are broken down by champion and formatting and language is fairly consistent across patch changes in relation to the type of change, allowing some automation to categorize the changes made to the game from the patches.

After manually examining the patch change text, we devised a taxonomy for classifying the individual patch changes, depicted in Figure 3. Each patch change is classified into one of the leaves in the taxonomy.

Broadly, there are three main types of patch changes:

- *Gameplay* – gameplay changes affect the champions and their interactions.
- *Bug fixes* – bug fixes are changes that correct inadvertent mistakes in the game (e.g., a bug fixed where interrupting an action would render the champion unable to cast spells).
- *Visual* - visual changes modify the look of the game for both the map and/or champions, including graphics, skins or other animations (e.g., new visual particles added when a spell is cast).

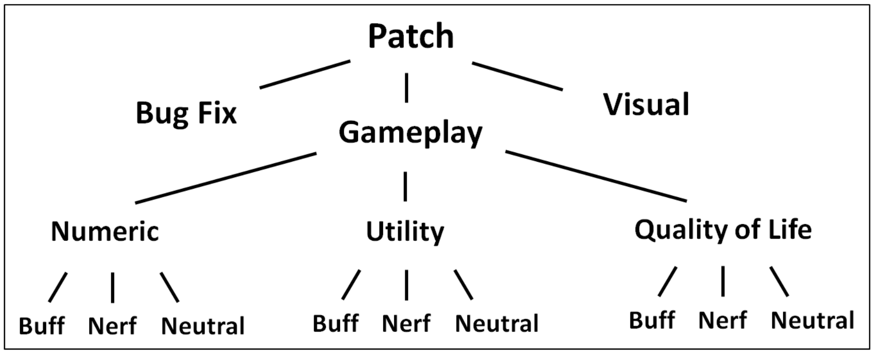

Figure 3: Patch Taxonomy. Leaves are final classification for each patch change.

For issues related to game balance, gameplay changes are of the most interest. Gameplay changes can be further categorized as:

- *Numeric* – Numeric changes are quantified alterations to game statistics for champions (e.g., amount of damage dealt per attack).
- *Utility* – Utility changes affect how a champion's ability interacts with other aspects of the game (e.g., an altered effect to an ability that slows the opponent when hit instead of damaging).
- *Quality of Life* – Quality of life changes affect the ease of use of a champion (e.g., an added visual indicator to better determine where a spell will hit).

Each gameplay change can be further identified based on the effect the change has in terms of the champion's relative strength:

- *Buff* – A buff increases the strength of a champion (e.g., base armor increased to 23 from 19).
- *Nerf* – A nerf decreases the strength of a champion (e.g., spell radius reduced to 300 from 350).
- *Neutral* – A neutral change is neither clearly a nerf nor a buff (e.g., base damage changed to level-1 150 / level-3 450 from level-1 100 / level-3 500 – the ability is being buffed at level 1 (going from 100 to 150 damage), but nerfed at level 3 (going from 500 to 450 damage)).

From visual inspection, gameplay changes are the most frequent, followed by visual updates and then bug fixes. For gameplay changes, numeric changes are the most frequent, followed by utility changes and quality of life changes.

For bug fix changes, the terms "fix(ed)" and "bug" are generally used, and for visual updates, the terms "visual," "animation," and "update(d)" are generally used. For gameplay changes, numeric changes can be often detected from words such as "increased," "reduced" and "modified" followed by a number – whether such a term is a nerf or buff depends upon the context. Quality of life changes do not use consistent language to help with identification for categorization. Utility changes occur most often when a champion is reworked (i.e., the abilities of the champion are significantly changed), but also lacks consistent language that can aid in categorization.

False negatives occur when a change is not detected and no categorization is made.

False positives occur when a change is detected but is incorrectly categorized.

True positives occur when a change is detected and correctly categorized.

We obtained all available patch notes from the League of Legends Wiki,[13], a community edited resource with information on each champion based on patch files released by Riot Games. For each champion, we extracted the patch change text, categorized it and stored it in a database for analysis. A combination of AutoIt[14] and Perl scripts were used to pull and parse the patch changes for each champion.

Categorization proceeded next using pre-determined keywords for automatic classification for bug fixes, visual changes and numeric gameplay changes. Automatic classification of quality of life and utility gameplay changes was not done as this greatly increased false positives. The resulting increase in false negatives was less detrimental since it was easier for a manual categorization step to notice when the automated system was unable to classify a change than it was to notice an incorrect classification. The final manual categorization step classified the remaining unclassified changes and corrected any mistakes made.

4 LoL Crawler Web Site

In order to allow for exploration of the game data – specifically, the champion win, pick and ban rates – and the patch changes, we implemented a Web site that provides for interactive exploration of our gathered data. The Web site:

http://lolcrawler.cs.wpi.edu/

is called LoL Crawler and allows users to pick a champion and then display champion game rates versus patch for that champion simultaneously and with links to champion patch changes. Lol Crawler is implemented in PHP, nodeJS[15] and C#. Graphing is done with HighCharts, a charting library written in Javascript.

A screenshot of the home page is displayed below in Figure 4. The home page provides brief overview of the site at the top and also presents a scrollable list of champions, shown with their images and names. This selection interface is similar to that used by players in the LoL client when choosing a champion before a game starts, so should be familiar to many users.

Selecting one of the images/names takes the user the data page for that champion, depicted in Figure 5. The very top of the page shows the champion name (e.g., "Darius"). The figure underneath shows a graph of the champion rates. The x-axis is the patch file and the y-axis is the rate (percent). Win rate, pick rate and ban rate are selectable on the right,

[13] http://leagueoflegends.wikia.com/wiki/Patch/

[14] https://www.autoitscript.com/site/

[15] https://nodejs.org/en/

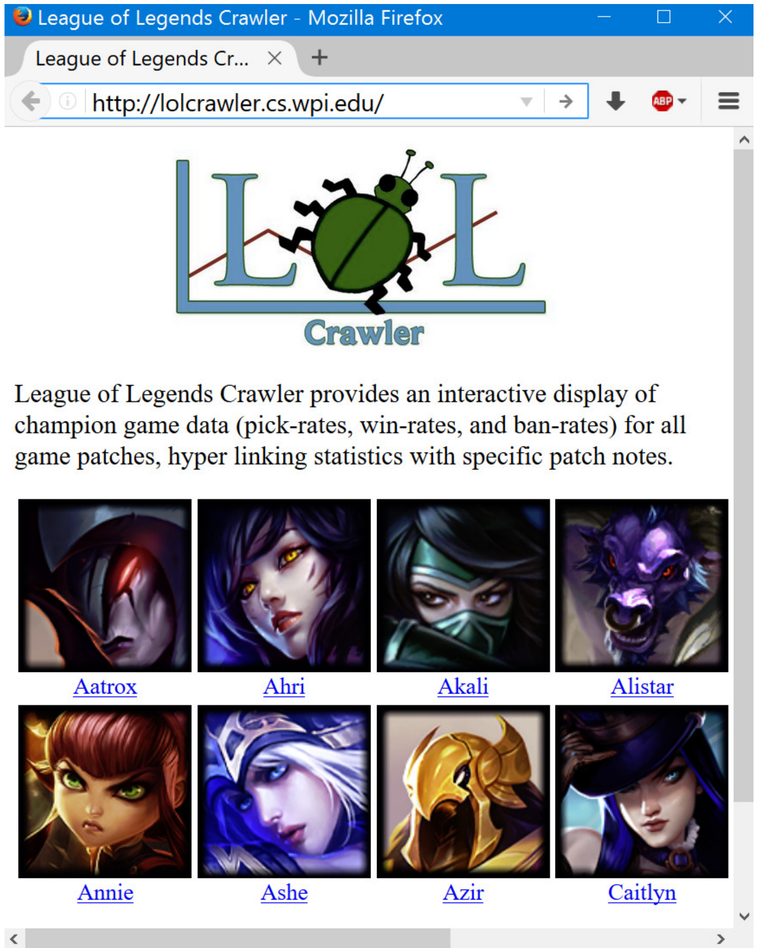

Figure 4: Home page from the LoL Crawler Web site.

toggling them on and off when selected (in Figure 5, they are all on). The x-axis is scaled to show only a limited number of patch changes to aid readability and is scrollable over the full range of patch files.

The bottom half of the page shows the patch changes, indicating the patch file number and the champion-specific patch changes each contains. Patch change categorization is also shown along with the accompanying text. Hovering over a data point in the graph pops up an information tooltip with the patch changes that correspond to that time. Clicking on a data point scrolls the patch change text below to the location of the corresponding changes.

5 Analysis

This section first analyzes the game data gathered through the Riot Games API (Section 5.1) and the patch data harvested in our classification process (Section 5.2), and then explores relationships between the two (Section 5.2).

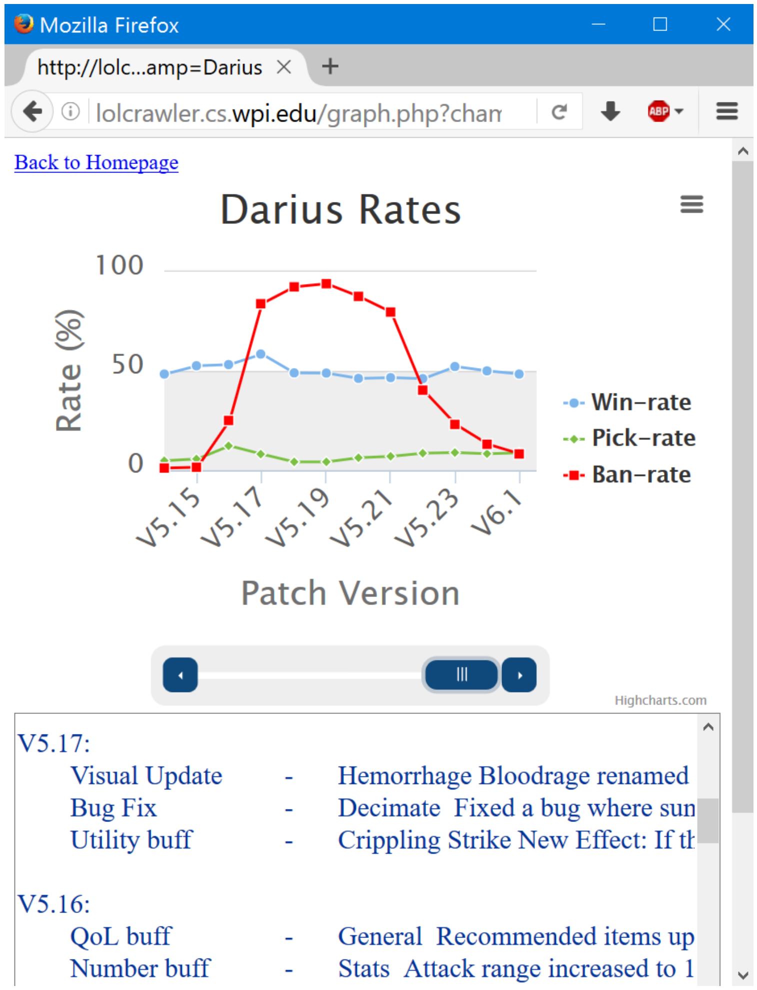

Figure 5: Champion data page from the LoL Crawler Web site.

5.1 Game Data

We applied our game sampling method (see Section 3.1) using the North American server, resulting in 465,000 ranked games from Season 4 to the start of Season 6 (v6.1) from the combined history of 11,000 players. The game data provided lists of the 10 unique champions picked in each game, which champions won and which lost, as well as the 6 champions that were banned each game. In total, the process yielded statistics on nearly 7.5 million champions.

From the data gathered, we computed the most fundamental statistics on the selection and then effectiveness of player choices – specifically, the win rate, pick rate and ban rate for each champion.

Figure 6 depicts a graph with the cumulative distribution functions for each rate, computed as a percentage calculated for each champion for each patch file. The x-axis is the rate percentage and the y-axis is the cumulative distribution of the rate. From the graph, the win rates are centered around 50%, which makes sense since in a given game half the

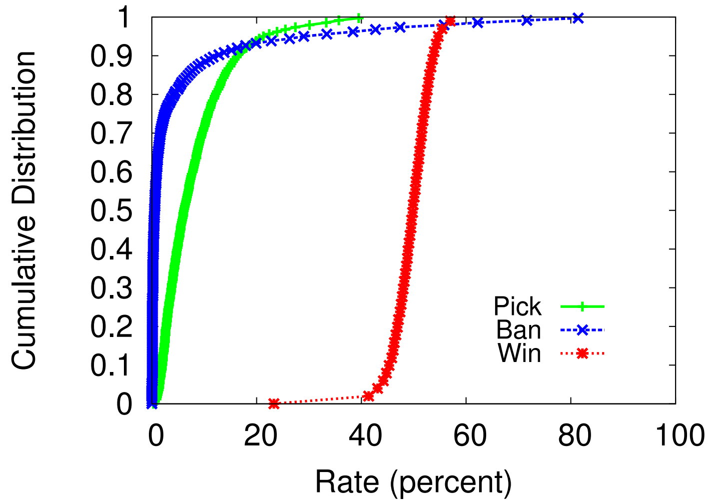

Figure 6: Win rates, pick rates and ban rates for all champions calculated for each patch file.

champions win and half lose. The individual champion win rates are not equivalent – if they were, all champions would have the they same win rate and the graph would show a vertical line at 50%. Instead, the bulk of the distribution is between 45-55%, with the "S" shape suggesting the values are normally distributed.

Figure 7 depicts a quantile-quantile (Q-Q) plot comparing the win rate distribution to a normal distribution. The x-axis is the normal distribution quantile (z-score) and the y-axis is the win rate value. The sorted win rate values are plotted as points, with the diagonal line being $y = x$. In a Q-Q plot, if the distributions are similar, the points will lie upon the line.

In the case of Figure 7, the data does appear to lie upon the line – i.e., the data looks normal – and the correlation coefficient is 0.991. However, the tails of the distribution do not fit a normal distribution as well as the rest of the distribution. The curves that deviate are because there are fewer win rates in the tails of the distribution than would appear in the tails if it were a normal distribution. Riot Games has most likely intentionally addresses win rate outliers by keeping champions from having significantly higher win rates (too strong) or significantly lower win rates (too weak) than the 50% average.

The pick rate distribution in Figure 6 is more skewed than the win rate distribution. The pick rates range from about 5% for half of the distribution to over 20% for top 2% of the distribution. The top 5 pick rates are for the champion Lucien for patch files 126 (v4.8), 127 (v4.9), 134 (v4.16), 135 (v4.17), and 164 (v6.1).

The ban rate distribution in Figure 6 has the most skew, where 70% of the champions have a ban rate of under 1%, but the top 2% of champions have a ban rate of over 40%. The top 5 ban rates are for the champions Jax for patch file 4.10, Sejuani for patch files 147 (v5.8) and (148) (v5.9) and Darius for patch files 157 (v5.18) and 158 (v5.19).

Table 2 provides the rate summary statistics.

LoL is a cooperative game where champions are designed to be played with a set role

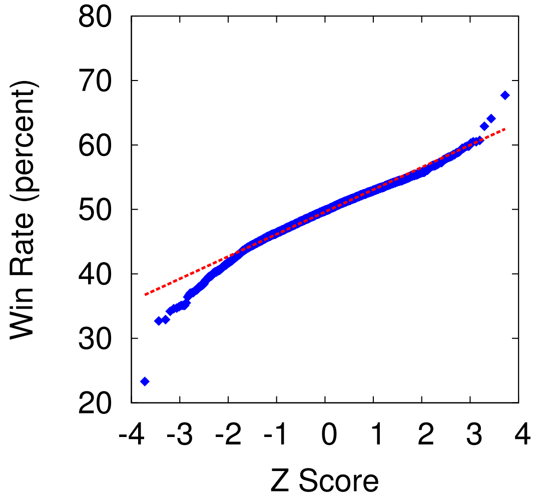

Figure 7: Visual normality test (Q-Q plot) for win rate distribution. Graph is square scatter plot of (sorted) win rates versus z score and dashed line is least squares linear regression.

Table 2: Summary Statistics of Rates

Statistic	**Win Rate**	**Pick Rate**	**Ban Rate**
Min	23.3 %	0.3 %	0.0 %
Median	49.8 %	6.1 %	0.5 %
Mean	49.6 %	8.1 %	4.9 %
Std Dev	3.4	6.8	12.4
Max	67.7 %	50.2 %	94 %

Table 3: Summary Statistics of Rates for Champion Roles

Role	**Win Rate**	**Pick Rate**	**Ban Rate**
Assassin	49.0 %	4.3 %	6.9 %
Fighter	50.9 %	9.0 %	5.8 %
Mage	49.7 %	7.5 %	3.8 %
Marksman	49.8 %	11.4 %	1.8 %
Support	50.3 %	10.0 %	6.1 %
Tank	50.3 %	4.6 %	3.8 %

in mind during combat (see Section 2.1). Table 3 provides the summary statistics for the rates from Figure 6 broken down into the designated champion roles. Win rates are near 50% for all champions, with Fighters slightly higher and Assassins slightly lower. Perhaps correspondingly, Assassins have the lowest pick rate. The Marksman has the highest pick rate and the lowest ban rate.

Given the competitive nature of ranked play (winning improves player ranking), it would

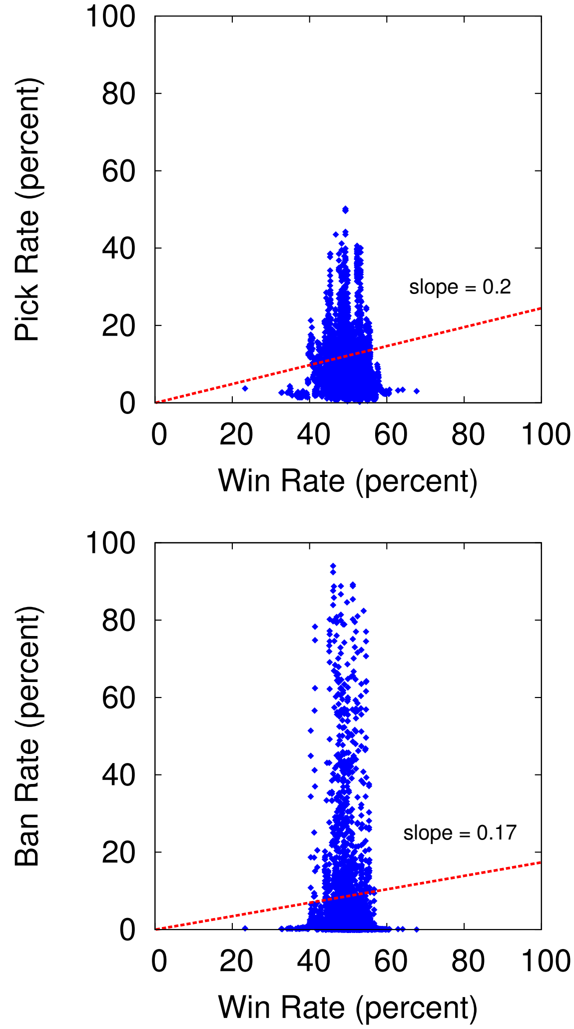

Figure 8: Pick rate versus win rate and ban rate versus win rate. Graphs are shown as square scatter plots and dashed lines are least squares linear regressions.

be natural for champions that win more often to be picked (and banned) more often.

Figure 8 depicts scatter plots of the pick rate (top graph) and the ban rate (bottom graph) versus the win rate. Each dot is the rate for one champion for one patch file. The dashed horizontal lines are least squares linear regressions. From the graphs, there is considerable variance in both the pick rate and the ban rate versus the win rate. However, the slopes of both linear regressions trend upward, suggesting a tendency for players to pick and ban champions that win more often.

5.2 Patch Data

In total, Riot Games has applied path files to LoL 164 times with 7710 total patch changes. The automatic part of our system classified 67.0% of all changes with a true positive rate of

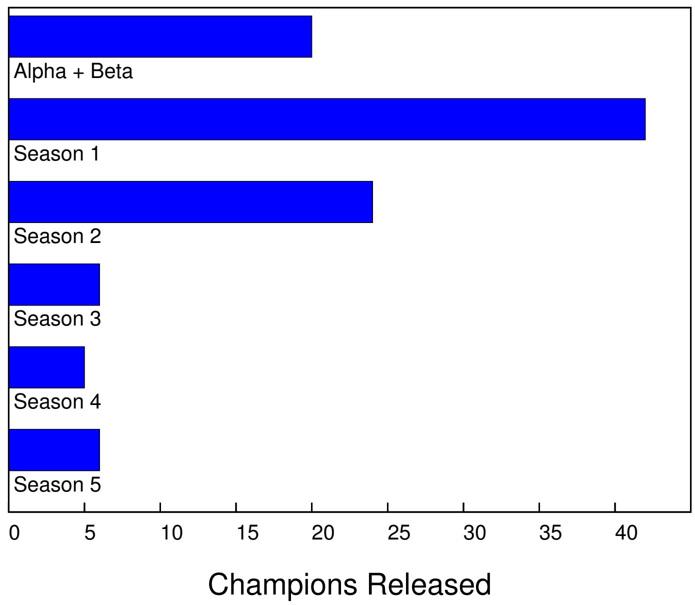

Figure 9: Champions released each season.

63.2%, a false negative rate of 33.0%, and a false positive rate of 3.8%. The precision and recall were 94.3% and 65.7% respectively. Manual inspection fixed the miss-classified patch changes and classified the remaining patch changes. No new false negatives or false positives were revealed and all true positives were confirmed.

LoL was in Alpha and Beta testing from February 2009 until its official release in July 2010. Since then, there has been approximately one competitive season each calendar year (i.e., Season 5 ended November 2015).

A major change in the gameplay added by patch files is additional content in the form of new champions. While no patch files after the alpha and beta seasons added more than one champion at a time, the number of champions released has varied widely with season (year). Figure 9 depicts a horizontal bar chart of the number of champions released each season. The y-axis is the season and the x-axis is the number of champions released. Early seasons had many more champions released, as often as about one champion every two weeks, with more recent seasons having released less than one champion every four weeks, on average. As of the time of our analysis,[16] there were 128 champions total.

Figure 10 depicts the total number of changes extracted from the patch files. The x-axis is the days since the release of the closed Beta, April 11, 2009.[17] The y-axis is the number of changes in each patch file. Visually, there is considerable variance in the number of changes per patch file, but no clear downward trend as might be expected from long-running software that has reached maturity. In fact, a least squares linear regressions has a slope of 0.000005, suggesting a very slight upward trend in the number of changes each patch file.

Figure 11 depicts a cumulative distribution function (CDF) of the top level – Bug Fix,

[16]February 2016
[17]Patch files 1-6 were for the closed Alpha and only have a year (2009).

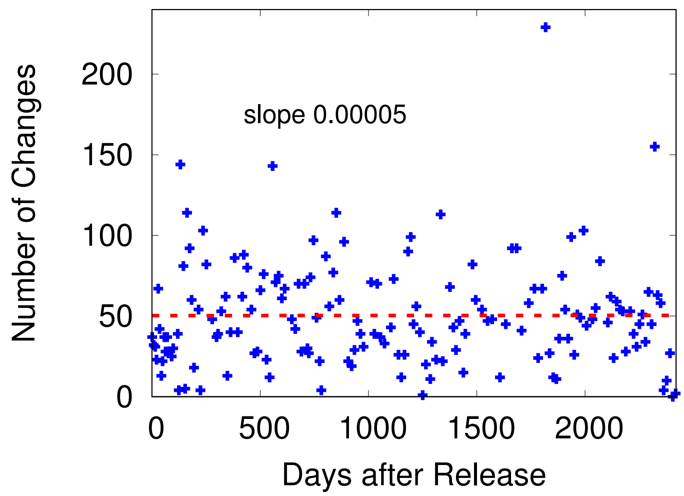

Figure 10: Total patch changes versus patch file.

Gameplay, Visual – patch changes to LoL from our taxonomy (see Figure 3). The x-axis is the number of patch changes per patch file and the y-axis is the cumulative distribution. There are three trendlines, one for each type of patch change. Bug fixes, the most common patch type in traditional software, are in the obvious minority with a median of only 2 fixes per patch. Visual changes to the game are only somewhat more common, with a median of 5 changes per patch file. Contrast that with gameplay changes where the distribution is significantly shifted to the right, with a median of 34 changes per patch file.

Figure 12 depicts a CDF of the second level gameplay – Numeric, Utility, Quality of Life – patch changes to LoL from our taxonomy (see Figure 3). The x-axis is the number of patch changes in a patch file and the y-axis is the cumulative distribution. There are three trendlines, one for each type of gameplay patch change. Quality of Life changes are the rarest, perhaps because these are the most significant in terms modifying how a champion is played, and have a median of 2 changes per patch file. Utility changes are more common with a median of 6 changes per patch file. Numeric changes are the most common, perhaps because they are the easiest to enact in terms of code, with a median of 26 changes per patch file.

Figure 13 depicts a CDF of the third level – Buff, Nerf, Neutral – patch changes to LoL from our taxonomy (see Figure 3). The x-axis is the number of patch changes in a patch file and the y-axis is the cumulative distribution. There are three trendlines, one for each type of third level patch change. Neutral changes are the least common, since most changes either strengthen or weaken a champion, with a median of 3 patch changes per patch file. Nerfs and buffs are fairly equally plentiful, with the distribution of nerfs just slightly to the right of that of buffs, a corresponding median of 18 and 16 patch changes per patch file, respectively.

Table 4 provides the patch data summary statistics.

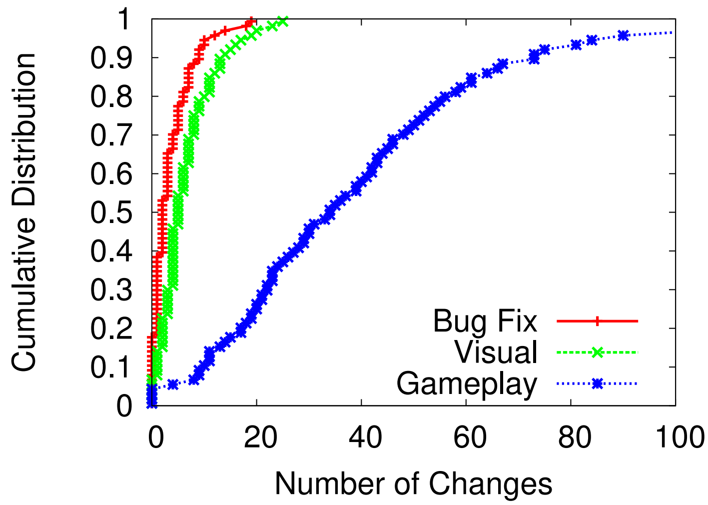

Figure 11: Cumulative distribution of taxonomy level 1 patch changes.

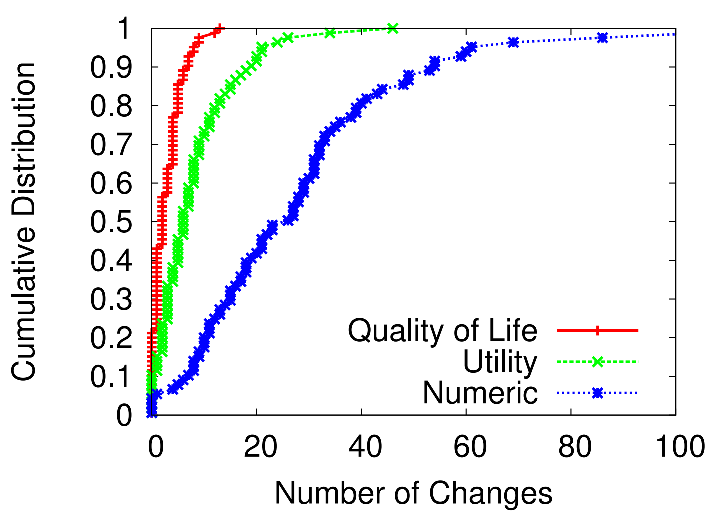

Figure 12: Cumulative distribution of taxonomy level 2 patch changes.

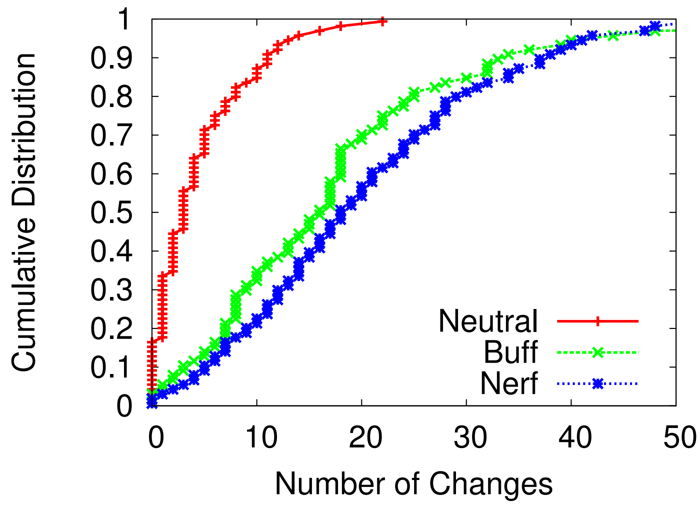

Figure 13: Cumulative distribution of taxonomy level 3 patch changes.

Table 4: Summary Statistics of Patch Data

	Number of Changes				
Type of Change	**Min**	**Median**	**Mean**	**Std dev**	**Max**
Gameplay	0	34	39	29	223
Visual	0	5	7	6	34
Bug fix	0	2	4	4	24
Numeric	0	26	28	23	184
Utility	0	6	8	8	46
Quality of Life	0	2	3	3	13
Buff	0	16	19	19	159
Nerf	0	18	19	12	57
Neutral	0	3	4	5	24

5.3 Combined

As mentioned in Section 5.1, win rate may intentionally have a restricted spread because Riot Games attempts to keep overall champion win rates as close to 50% as possible. Any win rate higher than 50% implies that the champion is giving the player an advantage (a dominant strategy) while any win rate lower than 50% implies the champion is giving the player a disadvantage (a dominated strategy). In order to maintain balance, Riot Games may choose to appropriately nerf or buff individual champions to keep win rates near 50%. Figure 14 depicts a scatter plot of the win rate against the number of patch changes for that champion. A best-fit quadratic regression for the data shows that as the win rate moves farther from 50% the number of patch changes increases for that champion in that patch file.

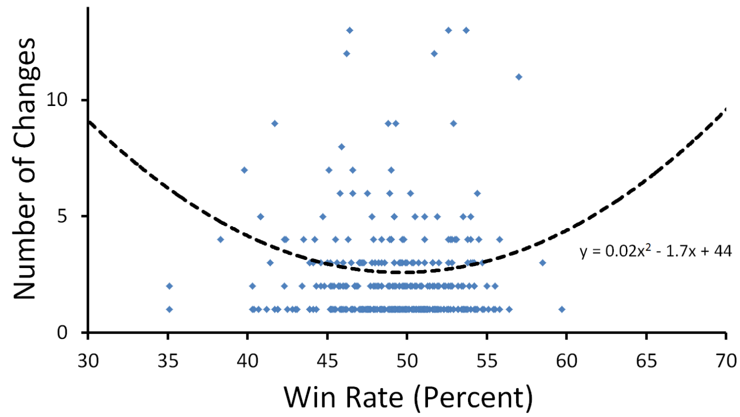

Figure 14: Number of patch changes versus win rate. Number of patch changes is the count of changes applied to a champion for a patch file and the win rate is computed after the patch changes are applied.

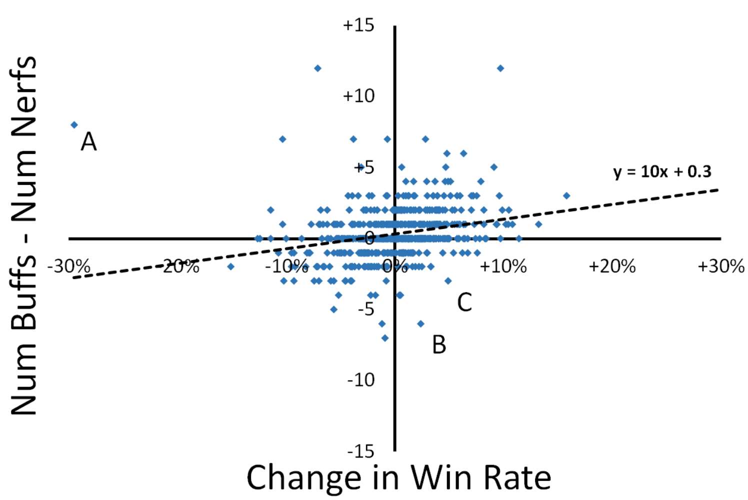

Figure 15: Buffs minus nerfs versus win rate. The count of buffs and nerfs are extracted from the patch changes and the win rate is computed after the patch changes are applied.

In addition, the expectation is that changes the increase the strength of the champion (i.e., buffs) increase the win rate for that champion, while changes that decrease the strength of a champion (i.e., nerfs) decrease the strength of that champion. Since for a given patch file, a champion may have some mix of nerfs and buffs, we compute the number of buffs minus the number of nerfs in each patch file. Positive values should increase the win rate of a champion, while negative values should decrease the win rate of a champion. Figure 15 depicts the result. The y-axis is the number of buffs minus number of nerfs. Positive y values mean the champion was strengthened by the set of patch changes in the patch file and negative y values mean the champion was weakened by the set of patch changes in the patch file. The x-axis is the subsequent change in win rate, as a percentage, after the changes have been applied. Positive x values mean the champion won more often after the patch changes and negative x values mean the champion lost more often.

From the graph, there appear to be more points in the upper right quadrant and lower left quadrant, as would be expected based on our reasoning above. A linear regression of the data is shown by the slanting line in the figure with an equation of $y = 10x + 0.3$. Roughly, this means every net increase to the strength of a champion results in a 10% increase in the champion's win rate and every net decrease to the strength of a champion results in a 10% increase in the champion's win rate. Note, however, there are many points in each quadrant, suggesting the quantification of buffs - nerfs does not always have a straightforward impact on win rate.

There are several outliers in Figure 15 that are interesting to note. Using our interactive LoL Crawler Website (see Section 4), the specific nerfs and buffs can be examined in conjunction with the subsequent win rate. That process is illustrated below.

The point marked A in Figure 15 represents the champion Urgot for patch v4.15, where he had a decrease in win rate of 29.5%, but the patch file contained 8 buffs. Before this patch, Urgot was an unpopular champion, with low win rates and pick rates. His pick rate doubled from 0.5% to 1% after this patch file due to a minor rework containing several buffs and a bug fix. The influx of players that chose him after the patch likely had not previously played Urgot regularly, who is an somewhat odd and difficult to play champion, most likely causing his low win rate to drop, despite the buffs.

Points B represents the champion Gangplank for patch file v5.14, where he received a large rework of his abilities. Overall many of the changes were small statistical nerfs to attributes such as armor and health that likely did not have a large effect on win rate. He also received many large neutral changes to his abilities that changed how they fundamentally worked. From the patch file wording, it was unknown whether these changes would be a net positive or negative, hence they were worded neutrally and classified as neutral. However, the new abilities turned out to be effective and synergistic with one another, leading to an increase in win rate.

Point C represents the champion Kalista for patch file v5.17, where she received three nerfs. Her win rate increased by 4.9% with no real explanation, as the nerfs were unambiguously negative. The most likely explanation is that changes to other champions caused the increase in win rate. For example, the same patch file made buffs to several Fighter champions that excel at killing Marksmen, such as Kalista, but Kalista is better than most other Marksmen at escaping, perhaps explaining why her win rate went up relatively to others, despite the nerfs.

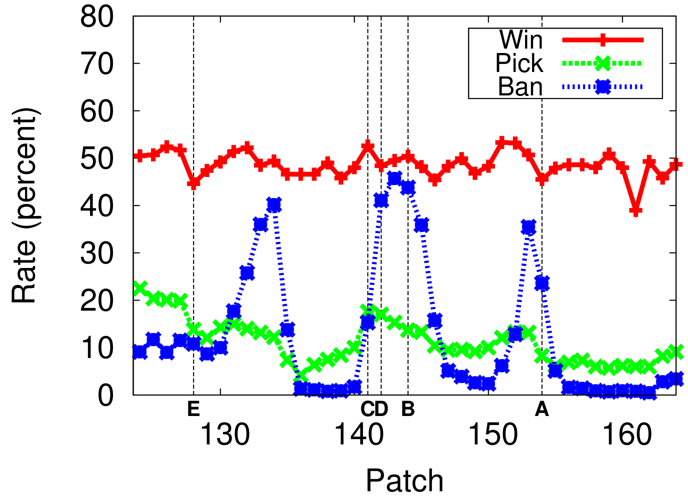

Figure 16: Rates for Nidalee versus patch file.

5.4 Case Study – Nidalee

"They will fear the wild." – Nidalee

To illustrate the interactive analysis enabled by our Web site, this section presents an in-depth case study of one champion – Nidalee – exploring correlations between game rates and categorized patch changes. In addition, interpretation of patch changes (hyperlinked on the Web from the rate graphs) provides insights not strictly available through categorization of patch change type alone.

Nidalee is primarily a ranged Assassin (see Section 2.1 for details on champion roles), but also can be played as a Fighter. Nidalee is also one of the oldest champions in the game. She was the 42nd champion, released in patch file v1.0.0.63[18] while LoL was still in closed beta. Thus, Nidalee has had a substantial history of changes both in balance and style as LoL has evolved. Her patch changes contain at least one of every patch classification in our taxonomy (Figure 3) and she went through a complete rework that comprehensively changed her abilities.

Figure 16 shows the win rates, pick rates, and ban rates for Nidalee versus patch file (shown for Season 4 onward). The x-axis is the patch file number and the y-axis is the rate, computed as a percent corresponding to data after each patch file. For example, a data point above patch file 154 (v5.15) means that rate data was taken after patch file 154 and before applying patch file 155.

A) A patch file that isolated the type of changes was patch file 154 (v5.15) with only numeric nerfs:

[18] December 17, 2009

Numeric nerf – Javelin Toss cast time increased to 0.25s from 0.125
Numeric nerf – Javelin Toss minimum damage reduced to 50/70/90/110/130 from 50/75/100/125/150
Numeric nerf – Javelin Toss maximum damage reduced to 150/210/270/330/390 from 150/225/300/375/450
Numeric nerf – Bushwhack on-cast vision radius reduced to 400

After this patch file, there is an observable drop in Nidalee's pick and ban rates since Nidalee was no longer as strong, and a corresponding drop in win rate.

B) As mentioned in Section 3.2, neutral changes have no obvious positive or negative effect – the exact impact depends upon the context. An example of this is in patch file 144 (v5.5):

Numeric neutral – Bushwhack damage changed to 40/80/120/160/200 (+20% AP) magic damage from 20/40/60/80/100 (+10/12/14/16/18 (+2% AP) health

where the base Bushwhack ability is increased with ability points (AP), but the bonus damage to low health champions is removed. This causes the attack to do less damage to low health enemies but more damage otherwise. Thus, there are game situations where this is a buff to the champion (e.g., normal attacks) but others were it is a nerf (e.g., attacks to wounded champions). The overall benefits depend upon how common each situation is and whether players can recognize and take advantage of them. In this case, it appears patch file 144 was decidedly a nerf as win rate immediately plummeted. Pick rate and ban rate followed the downward win rate more slowly since the impact of neutral changes are less obvious to players reading the patch text, only trending with win rate after being played for awhile.

C) Champion buffs of any kind usually come with increased pick and ban rates, especially utility and quality of life changes. While numeric changes just make a champion relatively stronger, utility changes often bring new features and quality of life changes that make a champion easier to use, leading to increased use. Patch file 141 (v5.2) illustrates this, with two changes – one quality of life buff and one utility buff:

QoL buff – General attack frame sped up slightly.
Utility buff – Prowl: Nidalee can now Hunt neutral monsters. Hunted neutral monsters are snared for 2 seconds.

The utility buff added a game feature and the quality of life buff made Nidalee feel more responsive, which led to an increase in pick rates and ban rates for a few patch files as players explored the changes. Win rate also responded with an increase as befitting a buff, but the overall impact on the game from these changes was not too significant.

D) Visual updates and bug fixes are generally not too influential on any of the three rates. While bug fixes can be influential if the bugs being fixed are strongly detrimental or beneficial to the champion, it is rare that bugs of this kind occur. Nidalee only has one isolated instance of a visual bug fix, patch file 142 (v5.3):

Bug Fix – Fixed a bug where Nidalee's cooldowns were not properly set when switching from Human to Cougar form.

In this patch, a bug was fixed where Nidalee's cooldown timers were sometimes set incorrectly when using one of her abilities (the ability to turn into a Cougar). While potentially significant (cooldown timers determine when an ability can be re-used), the relatively little change in the rates suggests overall negligible impact.

E) Nidalee underwent a rework in patch file 128 (v4.10), receiving 45 patch changes in one patch file: 22 buffs, 10 nerfs, 10 neutral changes, 2 visual updates, and 1 bug fix.[19] Such reworks often encourage more players to pick a champion since there are new gameplay elements to experience. However, some reworks simultaneously push away other players who prefer the champion's former features. For Nidalee and patch file 128, the latter case seemed to hold – she was a popular champion before the rework and even though the majority of the rework changes consisted of buffs, her pick rate decreased, possibly due to a dislike of the rework by those that played her regularly.

The win rate after a champion rework usually varies as both old and new players learn how to play the champion with the new changes. After Nidalee's rework, there was a significant drop in win rate similar to the drop in pick rate.

Ban rates are usually less volatile – ban rates before a rework mostly remain the same after the rework. It is rare for a champion's ban rate to rise or fall significantly with a rework alone unless the changes were clearly positive or negative. One example was for a rework of Darius in patch file 155 (v5.16), where his ban rate went from 1.4% in patch 153 (v5.14) to 94% (the highest ban rate we observed) in patch file 158 (v5.19). For Nidalee's rework, there was no significant change in ban rate – she was a popular ban prior to the rework and remained so after.

6 Conclusion

While traditional software patches typically only fixed bugs or tweaked performance, modern computer games often use software patches to adjust gameplay and even release more game content. League of Legends (LoL, Riot Games, 2009), one of the most popular games on the Internet, uses software patches not only to fix bugs, but also to add new player avatars (champions), in addition to changing the game balance as these avatars engage in combat.

Despite the importance of game balance to player satisfaction (Claypool et al., 2015), there has been little analysis of the effects of LoL patches on gameplay. Existing 3rd party Web sites allow players to examine champion statistics for the most recent LoL patches, but do not allow for inspection of patch effects over time. Prior research has more extensively analyzed the impact of software patches, but primarily for traditional software and in relation to software security. To the best of our knowledge, there has been no analysis and there are no 3rd party Web sites that allow for exploration of LoL patch changes in conjunction with the impact on player champion performance.

To analyze the impact of patching on League of Legends, our project gathered and analyzed over two years of LoL game data and over eight years of LoL patch data. Gathering game data required development of a methodology to obtain a random sample of LoL games from Riot Games and gathering patch data used a novel automatic classification technique to categorize patch changes from patch file text scraped from Web pages. The game data was deconstructed into the most fundamental game attributes for LoL players – champion

[19]Corresponding patch change text not shown here due to their length, but can be found at the Web site.

win rates, champion pick rates and champion ban rates. The patch data was categorized using a novel taxonomy that identifies patch changes based on their kind of change to the LoL champion and the relative change. In addition, we designed and developed a publicly available Web site[20] that allows for interactive exploration of game data and patch data for individual champions.

Analyzing game data for 128 unique champions played by more than 11,000 players in over 465,000 ranked games shows champion win rates are tightly clustered around 50%, normally distributed but with fewer with rates much higher or lower than 50% than would be expected. Pick rates and ban rates are significantly skewed, reflecting the disparate popularity of champions and perceived strength of opposing champions, respectively.

Analyzing patch data for more than 160 patch files with over 7700 patch changes shows gameplay changes dominate, being 5-10 times more numerous than bug fixes or visual changes to the game. LoL has added over 40 completely new champions since the first season. On average, LoL has about 75 patch changes to gameplay each month and only 4 bug fixes each month. Most (70%) of the gameplay changes are numeric increases (buffs) or numeric decreases (nerfs) to champion abilities, but there are also utility changes that make a champion easier to use and even quality of life changes that completely rework how a champion is played.

Analysis combining game data and patch data shows champions that have win rates further from 50% are patched most often, with buffs used to increase a champion's win rate and buffs used to decrease it. Interactive examination of game data and patch data illustrates the impact of neutral changes, visual updates and bug fixes as well as champion reworks. Use of the Web site, as well as the analysis in the paper, should be useful for LoL players and game designers that want to better understand how current and future patches affect champions and gameplay.

Future work could include study of other LoL game modes (e.g., Dominion, an alternate map and game objective) and additional analysis of other objective game data available through the Riot Games API, such as individual champion statistics (e.g., kills/deaths/assists) and in-game currency and items. Other possible future work includes analyzing patch data in relation to game data for other games – those of a similar genre to LoL are *Defense of the Ancients* (Valve, 2013) or *Heroes of the Storm* (Blizzard, 2015), but other game genres (e.g., first person shooter games) may use game patches differently.

References

Champion.gg - LoL Champion Stats, Builds, Runes, & Masteries. Online at: `http://champion.gg/`. Retrieved October 15, 2016.

Lol - League of Legends Summoner Stats & Champion Build Guides. Online at: `http://www.lolking.net/`. Retrieved October 15, 2016.

Patch. Online at: `http://leagueoflegends.wikia.com/wiki/Patch`. Retrieved November 18, 2016.

[20]`http://lolcrawler.cs.wpi.edu/`

Arora, A., Krishnan, R., Telang, R., and Yang, Y. (2010). An Empirical Analysis of Software Vendors' Patch Release Behavior: Impact of Vulnerability Disclosure. *Information Systems Resarch*, 21(1).

Claypool, M., Decelle, J., Hall, G., and O'Donnell, L. (2015). Surrender at 20? Matchmaking in League of Legends. In *Proceedings of the IEEE Games, Entertainment, Media Conference (GEM)*, Toronto, Canada.

E-Sports Earnings. Largest Overall Prize Pools in eSports. Online at: `http://www.esportsearnings.com/tournaments`. Retrieved June 22, 2016.

E-Sports Earnings. League of Legends 2014 World Championships. Online at: `http://www.esportsearnings.com/tournaments/7376-lol-2014-world-championship`.

E-Sports Earnings. League of Legends 2015 World Championships. Online at: `http://www.esportsearnings.com/tournaments/15149-lol-2015-world-championship`. Retrieved June 22, 2016.

Gkantsidis, C., Karagiannis, T., and VojnoviC, M. (2006). Planet Scale Software Updates. *ACM SIGCOMM Computer Communications Review*, 36(4):423–434.

Hullett, K., Nagappan, N., Schuh, E., and Hopson, J. (2011). Data Analytics for Game Development: NIER Track. In *Proceedings of the 33rd IEEE International Conference on Software Engineering (ICSE)*, Honolulu, Hawaii.

Jeppesen, L. B. (2004). Profiting from Innovative User Communities: How Firms Organize the Production of User Modifications in the Computer Games Industry. Technical report, Department of Industrial Economics and Strategy, Copenhagen Business School, Cophenhagen, Denmark.

Kow, Y. M. and Nardi, B. (2010). Culture and Creativity: World of Warcraft Modding in China and the U.S. In *Online Worlds: Convergence of the Real and the Virtual*, pages 21–41. Springer.

Lin, Z., Jiang, X., Xu, D., Mao, B., and Xie, L. (2007). AutoPaG: Towards Automated Software Patch Generation with Source Code Root Cause Identification and Repair. In *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 329–340, Singapore.

Lizardi, R. (2012). DLC: Perpetual Commodification of the Video Game. *Democratic Communiqué*, 25(1).

Nieborg, D. B. (2005). Am I Mod or Not? - an Analysis of First Person Shooter Modification Culture. In *Proceedings of the Creative Gamers Seminar - Exploring Participatory Culture in Gaming*, Helsinki, Finland.

Picture of Horse (2016). Choosing the Right Champion. Online at `https://support.riotgames.com/hc/en-us/articles/201752864-Choosing-the-Right-Champion`.

Raptr - The Offical Blog (2015). Most Played Games: November 2015. Online at: http://goo.gl/bVytCX. Retrieved June 22, 2016.

Riot Games. Our Games. Online at: http://www.riotgames.com/our-games. Retrieved June 22, 2016.

Schryen, G. (2009). A Comprehensive and Comparative Analysis of the Patching Behavior of Open Source and Closed Source Software Vendors. In *Proceedings of the IEEE Conference on IT Security Incident Management and IT Forensics (IMF)*, Stutgart, Germany.

Sotamaa, O. (2004). Playing It My Way? Mapping the Modder Agency. In *Proceedings of the Internet Research Conference*, volume 5, pages 19–22.

Webb, S. D. and Soh, S. (2007). Cheating in Networked Computer Games: A Review. In *Proceedings of the 2nd International Conference on Digital Interactive Media in Entertainment and Arts (DIMEA)*, pages 105–112, Perth, Australia.