Introduction

Oh My Zsh is an open source, delightful community-driven (with 1,300+ contributors) framework for managing your zsh configuration. Includes 200+ optional plugins (rails, git, OSX, hub, capistrano, brew, ant, php, python, etc), over 140 themes to spice up your morning, and an auto-update tool so that makes it easy to keep up with the latest updates from the community.

There are many types of shells. The default for Linux is bash. Although the function of bash is already very powerful, its hint function is not powerful enough, and the interface is relatively monotonous, it is not an ideal tool for programmer. Then zsh appeared but the configuration is too complicated. So Robby Russell started this open source project named Oh My Zsh. It is so powerful. It is compatible with bash and has a powerful history function. When using the or direction key to find historical commands, zsh supports limiting the search. Multiple terminal sessions share history. It can intelligently spell and correct, with various completion functions: path completion, command completion, command parameter completion, plug-in content completion, and more. It can intelligently jump and browse and jump through directories.

This chapter, our team aims to analyze and recover the architecture of this open source system Oh My Zsh. We start by first discussing its stakeholders and then takes on different viewpoints to analyze Oh My Zsh’s performance to recover its architecture.

Stakeholders

To start with an analysis of its architecture, we talk about stakeholders to Oh My Zsh as an open source and widely used project. The definition for stakeholder was given as: A stakeholder in a software architecture is a person, group, or entity with an interest in or concerns about the realization of the architecture. Different types of stakeholders have different requirements and influences on the project. As an open source project, the category boundaries of React's stakeholders may not be very clear. So we try to state the stakeholders in our understanding.

stakeholder roles

As it is stated in the definition, we can classify stakeholders as ten roles according to their roles and concerns: Acquirers, Assessors, Communicators, Developers, Maintainers, Suppliers, Support staff, System administrators, Testers and Users. According this classification, we will analyze the stakeholders as following:

Acquirers

Acquirers oversee the procurement of the system or product.
Oh My Zsh was started by @Robby Russell in Aug 23, 2009 and he lead his team to develop this program so far. @Erica Tafavoti is one of the developers in 2009, and now the Digital Marketing Manager of On My Zsh. She can represent the commercial interests of users in negotiating with third-party suppliers and can also be investor representation in this group if a specific external investment is required to fund the project.

Assessors

Assessors oversee the system's conformance to standards and legal regulation.
They don't arrange a special person to be responsible for the legal regulation. All of the members in the core team are responsible for checking whether the pull-requests meet the standards of the Code of Conduct. In this case, assessors come from Oh My Zsh's Core Team instead of external legal entities.

Communicators

Communicators explain the system to other stakeholders via its documentation and training materials.
The member in the core team supply a lot of materials to help users understand the using method of Oh My Zsh. We can learn these materials from https://ohmyz.sh/community.html. We can find a lot of articles and videos in the website. The team member Robby Russell, Rachel M.Carmena, Brad Parbs and some others provide the link of articals. Team member Karl Hadwen, AJ O'Neal, Cyril Mougel .etc provide the videos for users to learn. So the communicators come from Oh My Zsh's core team too.

Developers

Developers are those people who construct and deploy the system from specifications.
The early developer of the Oh My Zsh is @Robby Russell and his team worker, with the popularity of React developing, more and more community developers join this development. Now we have over 1350 developers to improve the functions of Oh My Zsh. And the earliest update is submitted 25 days ago by the core developers.

Maintainers

Maintainers manage the evolution of the system once operational.
In Oh My Zsh, the core teams work as maintainers. They discuss the development

direction of Oh My Zsh and they accept and review pull-requests. From GitHub contributors, we view the main contributors and find those programmers who work for Oh My Zsh core team: @mcornella@robbyrusell@apjanke@fred-o@ncanceill.etc

Suppliers

Suppliers build or supply the hardware, software or infrastructure on which the system will run.
Oh My Zsh is integrated with many plugins to enrich its function, here We just list the most important ones:
Git: It is an open source version control software developed by Linus Torvalds to help manage
Linux kernel development, which can handle project management efficiently and at high speed.
PostgreSQL: It is a very free-featured object-relational database management system. PostgreSQL supports most of the SQL standards and provides many other modern features such as complex queries, foreign keys, triggers, views, transactional integrity, multi-version concurrency control, and more. Similarly, PostgreSQL can be extended in many ways, such as by adding new data types, functions, operators, aggregate functions, index methods, procedural languages, and more.
Ruby: It is a simple and fast object-oriented (object-oriented programming) scripting language
Python: Python is a cross-platform computer programming language. An object-oriented, dynamically typed language originally designed to write automated scripts (shells) that are increasingly being used for stand-alone, large-scale project development as versions are continually updated and new language features are added.
Django: Django is an open source web application framework written in Python. The MTV framework mode is adopted, namely model M, view V and template T.
Docker: Docker is an open source application container engine that allows developers to package their applications and dependencies into a portable image and then publish it to any popular Linux or Windows machine for virtualization. Containers are completely sandboxed and do not have any interfaces to each other.
React: React is a tool used to build UI. Its design philosophy is extremely unique, revolutionary innovation, outstanding performance, and code logic is very simple.

Support staff

Support staff is those who provide support to users for the product or system when it is running.
Github gives the project the greatest support, since this is an open source project, github provides such a platform that viewers can download the system to use or submit their unique insights to the development team in order to improve the performance of the system. This is the key to make the project widely used and more

and more sound.

System administrators

System administrators run the system once it has been deployed.
This project is an open source project which can be download and used by everyone. The users administrate the whole system after the project was deployed. If the users are private user, they will administrate the system themselves. When come to team users or corporate users, it is more likely to allocate a specified person to act as the system administrator.

Testers

Since Oh My Zsh is an open-source library and has been posted on the Github, a huge number of programmers have the access to test it. Therefore it is tested by the staff of Oh My Zsh and the contributors on the Github. When bugs are found, they can send emails or issue them on the Github for the staff to solve.

Users

As we know, Oh My Zsh is a JavaScript Library to create a shell in the cmputer. Thus, every student or programmer who is a front-end engineer may find it convenient and easy to use and learn the programming of a shell.

@Kacey Cornell is one of the developers in 2009, and now the Project Manager of On My Zsh because of her ability to marry her creative side with management in an agecy setting. She is responsible for the management of the entire life cycle of Oh MyZsh, including staff assignment, project schedule, project quality control and so on.
Oh My Zsh was started by @Robby Russell in Aug 23, 2009. He is the VP Engineering/Partner @planetargon, Old-timer Ruby on Rails developer, host of Maintainable Software Podcast he/him/his, and of course, the creator of @ohmyzsh.

Quality attributes

The table below illustrates seven scenarios of Oh My Zsh quality attributes:

Table 2: Quality attributes analysis

Quality Attribute	Scenery	Stimulus	Stimulation	Product	Environment	Response
Availability	An unexpected thing happens	Inside the system or outside the system	Error or delay	Process	Normal operation	The system pauses and outputs diagnostic information
Mutability	Add a new function for the structure or perfect current system	Developers and other contributors	Further need for the functions of the structure	A better usable oh-my-zsh	During testing, designing and running	Change the part of the structure and test after changing
Usability	A zsh user is trying to learn to install and use oh-my-zsh	Zsh users	Need for knowing how to install and use oh-my-zsh	The oh-my-zsh system	During using or configuring	Provide detailed readme.md to introduce how to install and use oh-my-zsh.
Security	An attacker is trying to invade oh-my-zsh service on user's PC	Vicious attacker	An attempt to do something vicious by violating oh-my-zsh process	Safety verification	Oh-my-zsh process is under attack	Verify the access attempt and report to user
Testability	After a new part of the structure was developed	Developers and testers	A newly developed structure	A new version of the structure	During designing, developing and deploying	Take enough tests

Performance	A user downloads a plugin or theme through oh-my-zsh to apply on his zsh	Users	Download and application	The oh-my-zsh system	Normal operation	Deal with user’s command

Context View

The context view describes the relationships, dependencies, and interactions between the system and its environment (the people, systems, and external entities with which it interacts). This section examines oh-my-zsh’s scope, its dependencies on others and the interaction with other parties.

System scope & responsibilities

Oh My Zsh is a community-dirven command line tool, as its home page says, Oh My Zsh is a way of life.

> *Oh My Zsh is just a configuration wrapper for the zsh command line environment, but it doesn't provide a command line window, not a standalone APP. It is an extended toolset based on the zsh command line that provides rich extensions. In short, Oh My Zsh is not a substitute for a command-line tool, but complements them.*

Oh My Zsh functions are mainly to make command-line tool more beautiful and more efficient. It has the following three aspects of responsibilities in general:

- Provide various theme configurations
- Provide plugin mechanism
- Built-in convenient operation

External entities and interfaces

Below, these are elaborated upon and afterwards visualized in Figure 1.

- Written in Shell
- Unix-like operating systems (macOS or Linux) and Windows are supported.
- Supports many programming languages(Python, php, etc.)
- Robby Russel and more than 1350 contributors from the open source community.

- Users are individual developers.
- A GitHub repository filled with code, plugins and many issues is used to host the code base.
- Communication and support is provided via GitHub.
- The project is licensed under the MIT License, a permissive free software license originating at the Massachusetts Institute of Technology.

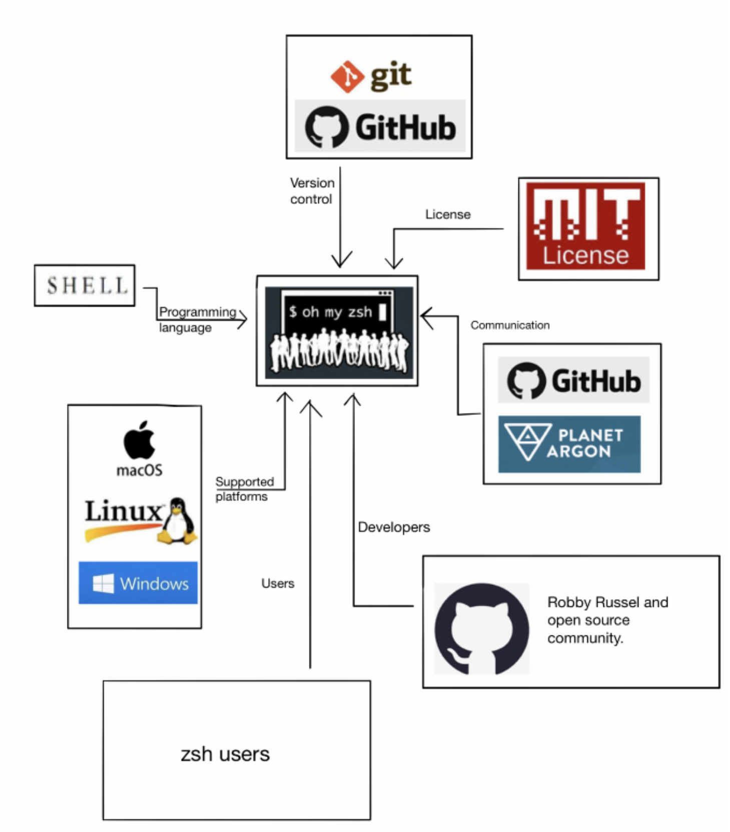


Figure 1: The Context View of oh-my-zsh

Function view：

According to Rozanski and Wood&apos‘s book：

> *Describes the system’s functional elements, their responsibilities, interfaces, and primary interactions. A Functional view is the cornerstone of most ADs and is ofte n the first part of the description that stakeholders try to read. It drives the shape of other system structures such as the information structure, concurrency structu re, deployment structure, and so on. It also has a significant impact on the syste m’s quality properties such as its ability to change, its ability to be secured, and it s runtime performance.*

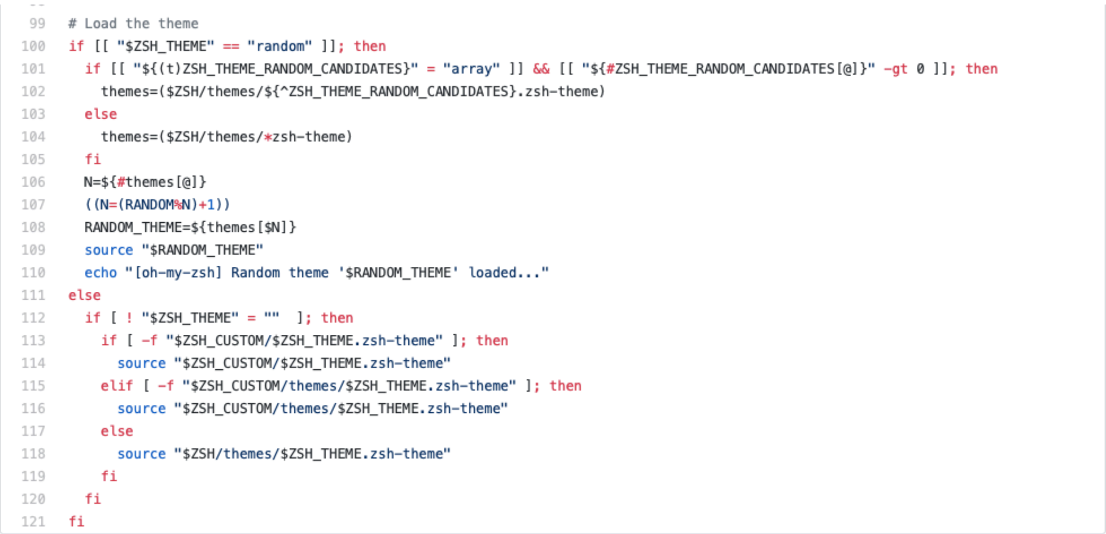
In this section, we will discuss the most important and unique functionalities of Oh-my-Zsh. Then we will also analyze extensibility of Oh-my-Zsh related to these functionalities. Since this is a project that focuses on UI, our functions are mainly focused on enhancing the user's sensory experience.

Functional capability

Function capability defines what the system is required to do and what it is not required to do. Since Oh My Zsh is a configuration wrapper for the zsh command line environment, it provides various of theme configurations and plugin mechanism which makes the shell's prompt function more powerful and the interface more dazzling. So the main functionalities that it needs to have coincided with that.

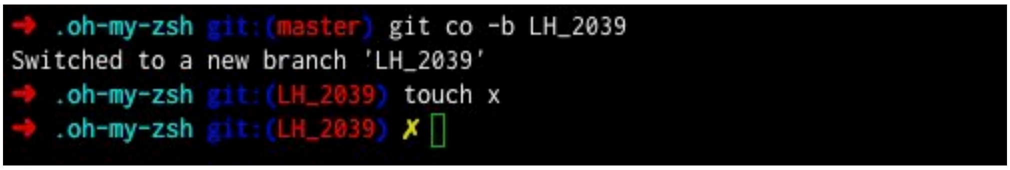
Theme

Oh-my-Zsh provides 140+ different themes, and users can change the theme on the shell at any time according to their own preferences. All of these themes have been continuously improved by the operator and stored in the library ' ~./oh-my-zsh/themes '. When the program starts running, the system uses the function in the file 'oh-my-zsh.sh'(details in the following picture) to implement the functions that call the theme in the library and display it on the user shell. Then the user can use the instruction 'ZSH_THEME=theme's name' to change the theme.
The code of loading theme:



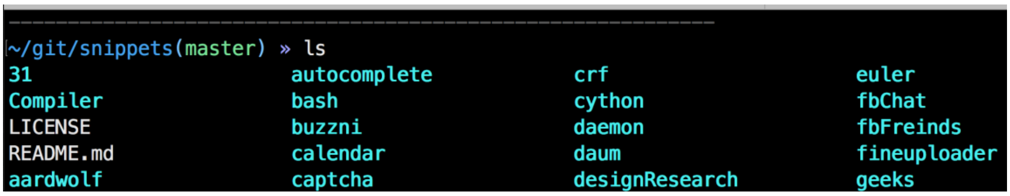
```
# Load the theme
if [[ "$ZSH_THEME" == "random" ]]; then
  if [[ "${(t)ZSH_THEME_RANDOM_CANDIDATES}" = "array" ]] && [[ "${#ZSH_THEME_RANDOM_CANDIDATES[@]}" -gt 0 ]]; then
    themes=($ZSH/themes/${^ZSH_THEME_RANDOM_CANDIDATES}.zsh-theme)
  else
    themes=($ZSH/themes/*zsh-theme)
  fi
  N=${#themes[@]}
  ((N=(RANDOM%N)+1))
  RANDOM_THEME=${themes[$N]}
  source "$RANDOM_THEME"
  echo "[oh-my-zsh] Random theme '$RANDOM_THEME' loaded..."
else
  if [ ! "$ZSH_THEME" = ""  ]; then
    if [ -f "$ZSH_CUSTOM/$ZSH_THEME.zsh-theme" ]; then
      source "$ZSH_CUSTOM/$ZSH_THEME.zsh-theme"
    elif [ -f "$ZSH_CUSTOM/themes/$ZSH_THEME.zsh-theme" ]; then
      source "$ZSH_CUSTOM/themes/$ZSH_THEME.zsh-theme"
    else
      source "$ZSH/themes/$ZSH_THEME.zsh-theme"
    fi
  fi
fi
```

Here list some frequently used themes:
Robbyrussell(the default that Robby uses)



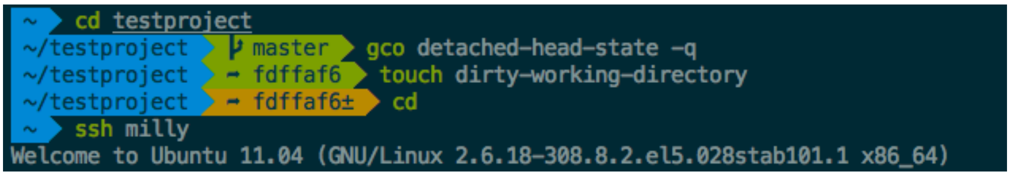
```
➜  .oh-my-zsh git:(master) git co -b LH_2039
Switched to a new branch 'LH_2039'
➜  .oh-my-zsh git:(LH_2039) touch x
➜  .oh-my-zsh git:(LH_2039) ✗ ▯
```

Af-magic(the rest of the themes, in alphabetical order)



```
------------------------------------------------------------
~/git/snippets(master) » ls
31                  autocomplete        crf                 euler
Compiler            bash                cython              fbChat
LICENSE             buzzni              daemon              fbFreinds
README.md           calendar            daum                fineuploader
aardwolf            captcha             designResearch      geeks
```

Agnoster(officicial repository)



```
 ~  cd testproject
 ~/testproject  master  gco detached-head-state -q
 ~/testproject  ➦ fdffaf6  touch dirty-working-directory
 ~/testproject  ➦ fdffaf6±  cd
 ~  ssh milly
Welcome to Ubuntu 11.04 (GNU/Linux 2.6.18-308.8.2.el5.028stab101.1 x86_64)
```

Plug-in

In order to enhance the user promotion of the project, oh-my-zsh uses a lot of built-in plug-ins to enrich his functions. The project puts all the plug-ins used by the program are put under the folder ‘ ~/.oh-my-zsh/plugins ’. In this document we draw a table to describe some of the popular plugins used in our program.

Name	Description
adb autocomplete plugin	Adds autocomplete options for all adb commands.
Django plugin	This plugin adds completion and hints for the Django Project’s commands and options.
Emoji plugin	Support for conveniently working with the Unicode emoji in Zsh.
Gradle plugin	The plugin adds completions and aliases for Gradle.
MySQL-Macports plugin	This plugin adds aliases for some of the commonly used MySQL commands when installed using MacPorts on macOS.
Python plugin	The plugin adds several aliases for useful python commands.
Ruby plugin	This plugin adds aliases for common commands used in dealing with Ruby and gem packages.
VS code	This plugin makes interaction between the command line and the code editor easier.

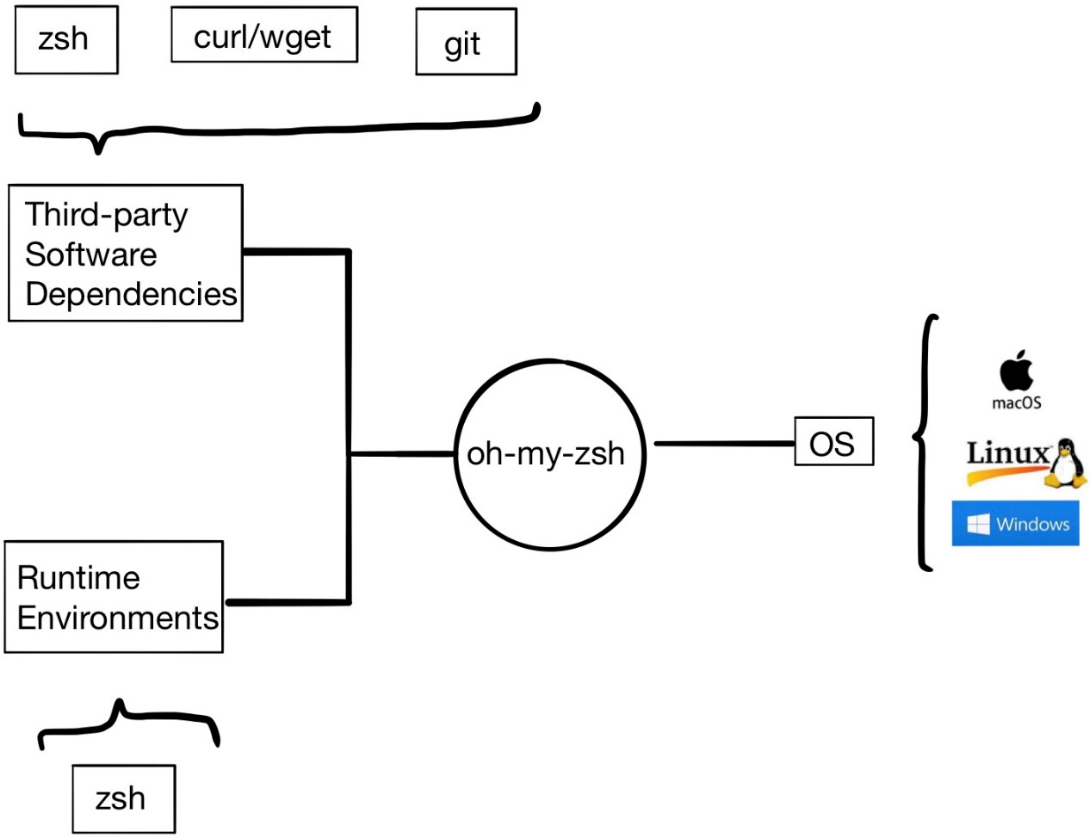
Extensibility

Since Oh My Zsh is just a configuration wrapper for the zsh command line environment and it is mainly to make command-line tool more beautiful and more efficient, its extensibility in terms of functionality is not so good. But in terms of its themes and plugins, it has a strong extensibility. We can customize our own theme and add our own plugins to enhance its functionality to make our zsh more powerful and beautiful.

We can just add a new file (ending in .zsh) in the custom/ directory to override any of the default behaviors. We can put them as a XYZ.plugin.zsh file in the custom/plugins/ directory and then enable this plugin if we have many functions that go well together. We can create a plugin of the same name in the custom/plugins/ directory and it will be loaded instead of the one in plugins/ to override the functionality of a plugin distributed with Oh My Zsh.

Deployment View

The deployment view describes the environment into which the system will be deployed, including the dependencies the system has on its runtime environment. As a framework for managing zsh configuration. Oh-my-zsh will be deployed on the computer system of the users.



Third-party Software Requirements

Since oh-my-zsh is a framework for managing your zsh configuration, it has the third-party software requirement zsh (v.4.3.9 or more recent).
What’s more , to enable its extra download-related functions, curl or wget should be

installed. In addition, git is also be required to be installed.

Specialist Knowledge

- Basic knowledge of using shell.
- Familiar with Unix-like OS.