



The Wikirate Project

▶ Annual Development Review 2

D6.6.2

Ethan McCutchen ▶ Decko Commons eV ▶ 10/23/2015

Dissemination level	Public
Contractual date of delivery	Month 24 September 2015
Actual date of delivery	Month 25 October 2015
Work package	WP6 WikiRate Architecture and Development
Deliverable number	D6.6.2 Annual Development Review 2
Type	Report
Approval status	Approved
Version	1.2
Number of pages	18
File name	D6_6_2-20151023_V1.2_Annual_Development_Review.docx

Abstract

This review of Year 2 technical progress surveys the current state of WikiRate.org, the code behind it, its technical support systems, and their APIs. The year featured the introduction of Metrics, which drove many platform and application improvements.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.



Co-funded by the European Union



History

Version	Date	Reason	Revised by
0.1	2015-10-09	First Draft	Ethan McCutchen
1.0	2015-10-16	Full Draft	Ethan McCutchen
1.1	2015-10-22	Polished Draft	Ethan McCutchen
1.2	2015-10-23	Incorporated Cambridge Suggestions	Ethan McCutchen

Author list

Organization	Name	Contact information
Decko Commons	Ethan McCutchen	ethan@decko.org
Cambridge	Richard Mills	rm747@cam.ac.uk



Executive Summary

The second year of The WikiRate Project involved major advances of the WikiRate codebase (T6.1), the Wagn platform (T6.2), the server technologies that support them (T6.3), and the mechanisms for connecting them all to the broader world (T6.4).

Metrics were a central strategic focus for WikiRate in grant year two, and a great portion of Task 6.1 (*Implementation of Features of WP2 and integration of WP5*) was devoted to their implementation. In order to create a context in which the Metrics Framework could thrive, we also revisited, redesigned, and updated nearly every major component of WikiRate.org.

The many *Improvements to Wagn platform* (Task 6.2) were conceived to support the new metrics functionality. Metric imports grew WikiRate's database by well over an order of magnitude, driving several performance enhancements, including optimized queries, query results cached in cards, and view caching. Other core improvements include Bootstrap integration, new card menus, upgraded file handling, customizable notifications, and gem reorganization.

WikiRate system administration (Task 6.3) was a major focus in year one, as we set up our initial technology stack. It will again be a major focus in year three as we move to a multi-server architecture. Year two featured highly valuable but less dramatic changes, such as the adoption of Semaphore for continuous integration testing, Yard for documentation, and Hound for enforcement of ruby style guidelines.

We have made continued progress on our *Application Programming Interface (API) and Plug-ins* (Task 6.4). The Metrics Framework pushed the boundaries of the mods API, and in response we refined its handling of subcards, action phases, errors, and edit conflicts.

Looking forward, we'll be supporting key marketing priorities in the coming year, including introducing ratings, promoting their adoption, engaging companies, and going multilingual.



Table of Contents

HISTORY	1
AUTHOR LIST	1
EXECUTIVE SUMMARY	2
TABLE OF CONTENTS	3
1 INTRODUCTION	4
2 TASK 6.1: IMPLEMENTATION OF FEATURES OF WP2 AND INTEGRATION OF WP5	5
2.1 WP2 FEATURES.....	5
2.1.1 PRIMARY PAGES.....	5
2.1.2 SECONDARY PAGES.....	7
2.1.3 OTHER CONTENT PATTERNS.....	8
2.2 WP5 INTEGRATION.....	9
3 TASK 6.2: IMPROVEMENTS TO WAGN PLATFORM	10
3.1 BOOTSTRAP.....	10
3.2 NEW MENUS.....	10
3.3 CARD QUERIES.....	10
3.4 PERFORMANCE.....	10
3.5 GEM ORGANIZATION.....	11
3.6 RAILS 4.....	11
3.7 FILE HANDLING.....	11
3.8 RENAMING.....	11
3.9 NOTIFICATIONS.....	12
3.10 DOCUMENTATION.....	12
4 TASK 6.3: WIKIRATE SYSTEM ADMINISTRATION	13
4.1 SERVER ARCHITECTURE.....	13
4.2 DEVELOPMENT ENVIRONMENTS.....	13
4.2 DEPLOYMENT AND BACKUP TOOLS.....	13
4.4 TESTING ENVIRONMENTS.....	14
5 TASK 6.4: APPLICATION PROGRAMMING INTERFACE (API) AND PLUG-INS	15
5.1 MODS API.....	15
5.2 RESTFUL WEB API.....	15
6 LOOKING FORWARD	16



1 Introduction

This document provides an overview of the current state of affairs of the four WP6 tasks:

- Task 6.1 *Implementation of Features of WP2 and integration of WP5*
- Task 6.2 *Improvements to Wagn platform*
- Task 6.3 *WikiRate system administration*
- Task 6.4 *Application Programming Interface (API) and Plug-ins*

It then presents a summary of the Development Plan for the year ahead.

This review is *not* intended as a chronological record of the efforts of the development team. Instead we outline the technical distance traveled in the second year of the grant. Therefore in the case of functionality that has undergone multiple rounds of improvements, this document will describe only the final state, not the incremental changes explored en route.

While the review does piece apart the deep integration of WikiRate.org and the Wagn framework, it does *not* make comparable efforts to piece apart the deep collaborations of the Decko Commons and Cambridge development teams. In general, the University of Cambridge focuses most heavily on Task 6.1 while Decko Commons focuses more heavily on the other three tasks. But both partners have contributed to most of the functionality that we've built, and pair coding and pair design sessions with representatives of both consortium partners present are routine.



2 Task 6.1: Implementation of Features of WP2 and integration of WP5

WikiRate.org is an instance of the Wagn wiki framework. Enhancements to the Wagn framework are discussed in Chapter 3; this chapter specifically reviews this year’s improvements to the WikiRate instance. These enhancements fall into one of two technological camps:

1. Code customization through the Wagn Mod API
2. Data configuration via the WikiRate.org site itself

As we survey the WikiRate features, we will regularly refer to this distinction as “mod code” vs. “wagneering” respectively.

The code customizations for WikiRate.org are publicly stored, managed, and discussed at <https://github.com/wagn/wikirate>. Database configurations are publicly viewable as cards at WikiRate.org.

2.1 WP2 Features

In this chapter we will first explore what has been implemented for all the major content types in terms of code customization and data configuration. Then we will explore some enhanced functionality shared across these types.

2.1.1 Primary Pages

At present, WikiRate.org features three *primary* content types: Companies, Metrics, and Topics. Each of these three is introduced on the homepage and among our first conversion goals is enticing users to explore these pages.

2.1.1.1 Home Page

WikiRate’s home page has been through several iterations as we attempt to find how best to introduce our core concepts in a manner that sets visitors on a path to deeper engagement.

Previous versions attempted to convey how *nearly all* of WikiRate’s many content types interact. At the time of writing, the homepage presents *just two* main content types: Companies and Metrics. However, based on feedback from our user base, we have decided to add a third – Topics – which we project to be deployed by the time of the annual review.

The homepage also features simpler, cleaner language about WikiRate’s mission and vision as well as, of course, the new logo and typology introduced in year two. User testing and feedback thus far suggests that the new homepage provides a considerably gentler entry into WikiRate.org while successfully conveying core concepts.

While the homepage does not require any custom ruby code, it does make use of JavaScript libraries (added by wagneering)



2.1.1.2 Companies

As WikiRate's central subject matter, Company pages have received and will continue to receive abundant attention.

Last year their core materials were lists of Overviews; now they strongly feature Metrics. Overviews, Notes, and Sources are readily navigable in tabs. At the time of the interim review, metric values could be vertically expanded to give a little inline detail; we have since reorganized the page to give a much more expansive half-page view (or full-page, in the case of mobile) of metric values.

The most code-intensive portion of the page is the handling of “important to me” – which allows users to drag metrics that they consider important to the top of the page; these metrics will then retain this prominence on all company pages. This is also the component that causes the greatest performance challenges, because it looks different to different users and therefore cannot be cached. Solutions for this challenge are outlined in D7.7.5 (*Interim Technical Evaluation 2*).

Among the most pressing remaining needs for company pages is the ability to filter metrics dynamically, including the capacity to show metrics missing values.

2.1.1.3 Metrics (and Metric Values)

Metrics were newly introduced in year two. Every metric has a compound name in the form *Metric Designer+Metric Label*. This insures that there is not a land grab dynamic around label names; metrics by different designers can use the same label.

The layout of Metric pages is designed to mirror the company page layout; companies list metrics, and metrics list companies.

Metric values are created through one of three means:

1. Direct form editing
2. Standard bulk import from CSV
3. Customized bulk import via Wagn's ruby API.

For each metric value the following cards also exist:

- *Metric+Company* – refers to all associated values
- *Metric+Company+Year* – the data for a single year
- *Metric+Company+Year+Value* – the actual numerical or categorical value for a given year
- *Metric+Company+Year+Source* – refers to source card for that value

Considerable design and development has also gone into facilitation of metric values navigation and discussion, including timelines and multi-perspectival summary views.

As with Metrics on Company pages, Companies on Metrics pages will require more sophisticated sorting and filtering, which we plan to introduce in coming months.

Currently under development are dynamically calculated Metrics, including Scores (in which metric values are normalized to a 0 to 10 scale) and Ratings (weighted averages of those Scores).



2.1.1.4 Topics / Tags

Our most significant changes to Topic handling are in progress at the time of writing.

Early feedback from the new Metric focus was that it was too difficult to navigate the site by interest area. WikiRate had two main mechanisms for this: Topics and Tags. Topics were predominantly for framing “Overviews”, whereas tags were for pretty much anything else.

We are now merging these two concepts such that Metrics, Notes, Sources, and Projects can be tagged with any Topic, which are no longer limited to a narrow array of sustainability issues. A Topic can then be an event, a theme, or anything else anyone might choose to tag. Any topic can be used for Overviews, though we expect that most will not. The more common usage will be organizing related Metrics, Notes, Sources, and Projects.

These new general-purpose Topics will be highly useful for helping users to navigate to content of interest. For those browsing WikiRate content, topic prominence will be determined by voting and activity.

2.1.2 Secondary Pages

As visitors explore WikiRate more deeply, they will encounter several other types beyond the primary pages outlined in section 2.1.1.

Overviews (formerly Articles) and Notes (formerly Claims) and Sources, for example, emerge when exploring WikiRate’s now-less-prominent qualitative side. Initiatives and Projects offer means for organizing activities, and User contributions appear throughout the site.

2.1.2.1 Sources

In year one, Sources were solely for citation in Notes. Now they are also cited by simple Metric Values. (A Source is required in both cases.) . Both Metric Values and Notes can cite multiple Sources.

In year two we expanded the kinds of Sources that WikiRate supports beyond url-based sources to include file sourcing and directly edited Sources.

We also completely overhauled the source interface to provide a more powerful, streamlined, and dynamic means of interacting with source material. It is now easy to add notes and metric values from source pages, to view existing notes and metric values that cite the source, to review and edit source metadata, or to hide all of the above and focus on scrutinizing the source itself.

It’s also worth noting that, while currently considered secondary material, the only button on the site header is the “Add Source” button. It is given this prominence because of the expectation that users reading a relevant source may think of WikiRate and wish to submit the Source immediately upon arrival to the site.

2.1.2.2 Contribution Pages (Users/Accounts)

In the near future, we will be adding support for corporate user accounts – accounts directly associated with Company cards. Unlike individual accounts, corporate accounts must be verified (initially by WikiRate eV), and only contributions from these accounts will be treated as official company responses. Individuals will be required to disclose formal relationships with Companies, but they will not be able to speak directly as company representatives through individual accounts.



While company accounts are not yet supported, companies can already be represented as Metric Designers. Therefore in the process redesigning our User Profile pages, we created company contribution pages, which are linked to from Companies that have designed metrics.

It's worth clarifying that WikiRate only makes functional distinctions for these two kinds of entities: Companies (which can be rated) and Individuals (which can't). While we will provide metadata about companies' legal status, place of formation, etc, these factors will not affect site capabilities.

The new contribution pages are organized to make it easy to navigate all of a user's contributions and to allow users to "spotlight" the contributions they consider most significant or most want to promote.

We also completed major improvements to "following" content early in year two; this content is highlighted prominently on contribution pages, as is recent activity.

2.1.2.3 Overviews

Given the decision to emphasize quantitative (Metrics) over qualitative content in year two (see D7.7,2), Overviews (formerly "Articles") got relatively light attention.

However, there were some improvements made to Overviews, such as immediately updating the list of cited/citable Notes whenever a citation is added to or removed from an Overview.

2.1.2.4 Notes

Similarly Notes (formerly "Claims") were deemphasized in year two, but Note pages were updated in keeping with stylistic upgrades and are very prominently linked to from Company pages.

Notes will soon be able to cite Metric Values as well as other Notes, so that they can be used for higher-level analyses and/or meta-level observations about site data and its usage.

2.1.2.5 Initiatives and Projects

Initiatives and Projects, two related structures for organizing user activity around key content themes, were introduced and honed in Year 2. Initiatives are large-scale undertakings associated with lists of companies. Projects are smaller-scale undertakings associated with a given Initiative. A given Project can be associated with one or more Metrics.

Both Initiatives and Projects are implemented entirely through Wagneering; as of yet there is no custom code required.

2.1.3 Other Content Patterns

2.1.3.1 Voting

Early in year two we completed deployment of our voting functionality. It is now possible to vote on Metrics, Notes, Topics, and Sources.

In addition to the conventional up-down voting interface, dragging a Metrics or Topics into the "Important to Me" section registers a vote for that item (see D4.4.1 for further information).



Metric voting will soon assume additional significance, as metric votes will be weighed into the WRIT (WikiRate Index of Transparency) scores. Metrics with more important metrics (ie, metrics with more net upvotes) will be weighed more heavily in measuring a company's transparency.

2.1.3.2 Styling and Responsiveness

As detailed below in Chapter 3, the Wagn platform was upgraded to use the Bootstrap platform in year 2. Bootstrap's mobile friendly patterns became central to WikiRate's styling in year 2, and nearly all aspects of the site were updated to use bootstrap components.

2.2 WP5 Integration

WikiRate's source suggestion service was disabled in year two because of the decisions (a) to de-emphasize qualitative content in year two, (b) to redirect CERTH resources towards preparing metric data for WikiRate import, and (c) that the initial rounds of source suggestions were not yet sufficiently reliable.

In the short term, the value added from CERTH's work on metric import remains high, so we do not expect to redirect those resources back to source suggestions or other functionality originally proposed as part of WP5.

In the medium-to-long term, however, we do hope to see the source suggestions improved and redeployed, and to integrate their work on other WP5 tasks.



3 Task 6.2: Improvements to Wagn platform

We have continued to implement drastic improvements to Wagn this year, and its community of users is growing. The Wagn ruby gem introduced last year has now been downloaded over 16,000 times.

3.1 Bootstrap

Bootstrap is a popular HTML, CSS, and JavaScript framework. This year it was integrated into the Wagn platform to support a host of new responsive (mobile friendly) behaviors.

This integration allowed us to upgrade almost all standard interface patterns on Wagn decks, including our iconography. We also now ship with 16 built-in bootstrap themes, and thousands more are available online.

3.2 New menus

One of the most significant interface upgrades is our new card menu, which makes heavy use of Bootstrap patterns. The new functionality better organizes all of Wagn's core functionality by more clearly emphasizing critical functions, better grouping more advanced features, and supporting discoverability through iconography.

The new menus also introduce new functionality, including view refreshing, menu pinning, grouped rules, and nest editing.

3.3 Card Queries

Card::Query is a core ruby Class in the *card* gem and plays the role of managing all WQL (Wagn Query Language) queries. Its primary function is to return lists of cards by generating and executing SQL queries.

As WikiRate's database grew from 20,000 cards to over 500,000, it became increasingly clear that Wagn's initial SQL strategy of translating nested card queries into nested SQL selects (via subselects) was sacrificing performance with scale, so Card::Query was rewritten to generate more traditional (and more concise) table joins. This has dramatically improved database performance and has led to considerably faster load times.

The new code also has a cleaner separation of query interpretation and SQL generation, such that it should now be much simpler to port Wagn to a NoSQL database if this proves advisable. As noted in D7.7.5 (*Interim Technical Evaluation 2*), this inquiry will be given more attention in the first calendar year quarter of 2016.

3.4 Performance

In addition to Card Query optimizations, several other improvements were made to Wagn in the name of improving site performance, particularly on WikiRate.org.

To facilitate these optimizations, we implemented a performance logging system that helps make clear how server time is being used so that areas of inefficiency may be identified.



Many of the related improvements came from addressing narrow code problems. But the larger improvements have come from enhancements to our caching systems. This includes a nascent view caching system, suggested improvements for which are detailed in D7.7.5 (*Interim Technical Evaluation 2*).

It also includes a new search result caching system, which in its initial implementation has been abstracted to function with a narrow but common set of searches.

3.5 Gem Organization

Last year we released Wagn for the first time as a ruby gem. This year the wagn code has been further modularized, and the wagn gem now requires two other gems from the same github repository: card and decko-rails.

The new modularity, achieved with considerable help from volunteer coders, is designed to make it possible to support using Wagn as a Ruby on Rails plugin. This use case should make Wagn appealing to a wider audience of ruby coders. While the use case is not yet fully available, the core code improvements are in place, and all current sites, including WikiRate, make use of the new gem suite.

3.6 Rails 4

In order to stay up to date with Rails code libraries and take advantage of new functionality such as ActiveJobs (supports delayed event processing, which helped improve card action performance by delaying follower notifications until after the initial server response), we upgraded Wagn from Rails 3 to Rails 4.

3.7 File Handling

In the process of the Rails 4 upgrade, we began experience serious bugs with Wagn's file handling library, called Paperclip. This drove us to carry out a long-planned improvement to use another library, CarrierWave, and update our file representation. This provided the following benefits:

- File uploads now start immediately upon selection – they previously only began when a form was completed and submitted
- Cards can now can have two files nested in one form, previously not allowed
- We will soon be able to support permission shortcuts for publicly viewable cards
- The new implementation is more idiomatic, and therefore will (a) require less maintenance and (b) prove easier to extend, for example with visualizations.

3.8 Renaming

A popular Wagn feature is that, when updating a card name, you can automatically update references to that card. This auto updating, however, did not include updates to cards directly referred to in WQL searches. For example,



if you were to rename “Claim” to “Note”, then a search for all Claim cards would not be updated to search for all Note cards.

This was formerly impossible because name references within search cards were not tracked in our reference system. We addressed this need, however, and now this renaming pattern works. This change was driven by the decision rename several core WikiRate cards as follows:

1. Claims to Notes,
2. Articles to Overviews, and
3. Campaigns to Initiatives

Because WikiRate’s card content included many thousands of tracked references to these names, the automated renaming drastically reduced the complexity of this task, and it also will make it much easier for other Wagn projects to rename their core structures as needed.

3.9 Notifications

We completed a thorough overhaul of our notifications system at the beginning of year two, adding great flexibility to which cards can be followed and how notification emails can be configured. In fact, all Wagn emails are now configurable through a unified system.

3.10 Documentation

This year we started using Yard, a ruby documentation framework, to enhance our documentation of Wagn code. We have made substantial progress in framing and explaining core classes and will continue to do so in advance of Wagn’s rebranding as “Decko”. Wagn (and Card) docs are now available on rubydoc.info.



4 Task 6.3: WikiRate system administration

4.1 Server Architecture

Our server architecture is largely unchanged from year 1. As discussed in D7.7.5 (Interim Technical Evaluation 2), this architecture is currently appropriate, but we plan to upgrade to a multi-server architecture in advance of more intensive promotion of our coming ratings features.

WikiRate.org is currently provided by a single Hetzner EX40 dedicated root server as follows:

- high-performance Intel® Core i7-4770 Quad-Core processor
- 32 GB DDR3 RAM.
- two 2 TB SATA 6 Gb/s

The same server is used for our internal documentation site (also using Wagn) and our public project site (wikirate.eu).

The development server is identical. It is used for testing, staging and demoing WikiRate changes. Both servers are running:

- Ubuntu 12.04.4 LTS
- Ruby 2.1.5
- Mysql 5.5.35
- Apache 2.2.22
- Passenger 4.0.29

4.2 Development Environments

As in year one, developers at Grass Commons and Cambridge developed primarily on laptops and then coordinate code changes with the rest of the development team through GitHub pull requests.

Each development team member also maintains an open Slack window for WikiRate, which integrates:

- Instant messages and group chats with team members (including nontechnical team members)
- GitHub updates and pull request notifications
- Updates to PivotalTracker stories
- Server Error notifications from Airbrake
- Notifications regarding automated testing from Semaphore

All developers continue to work collegially and to share resources openly. Daily meetings (standups) assure consistent coordination, and biannual developer retreats are used for intensive planning, designing, developing, and team building.

4.2 Deployment and Backup Tools



We currently use Capistrano to manage all our deployments to both the development and the production server. With Capistrano we can run full updates with a single command that manages:

- WikiRate code updates and data migrations
- Wagn code updates and data migrations
- Updates to dependent libraries
- Clearing tmp directories
- Restarting Passenger

We have also created Capistrano commands to:

- generate backups
- refresh the development copy of WikiRate.org from a backup of the production site
- copy and install a backup to a local development environment

Additional daily backups are generated on the production server and copied to the development server using cron jobs.

While authentication details are naturally kept private, all Capistrano commands are made public in WikiRate's github repository for reuse/modification in comparable environments.

4.4 Testing Environments

Both Wagn and WikiRate are tested using Rspec and Cucumber; WikiRate just added the Cucumber feature testing in year 2.

The most significant recent addition to our testing repertoire is Semaphore, a continuous integration testing tool that automatically runs applicable test suites whenever pull requests are submitted for Wagn or WikiRate.

Use of Semaphore has significantly reduced the number of bugs reported per deployment.



5 Task 6.4: Application Programming Interface (API) and Plug-ins

Wagn features two primary API's: the Mods API and the RESTful Web API. The former is the means by which Wagn's functionality can be altered or extended, and the latter is the means of interacting with Wagn data over the web.

5.1 Mods API

Wagn's mods API is the primary means by which functionality *not* distributed in default Wagn instances is created or customized for WikiRate.org.

Because *everything is a card*, essentially every Wagn mod involves changing how cards look or behave. This principle makes the API *simple*; what makes it *powerful* is its organization into Sets. The basic idea is that a Set is a configurable group of cards. A Set can be *all cards*, *a single card*, or something in between. The basic file in a mod is a *set module*, which, in essence, allows developers to customize how cards in a given Set are transformed (via Model methods) or displayed (via Format methods).

Key improvements to the Mods API in year two include:

- The handling of “subcards” has been honed significantly, allowing for nested bulk actions on cards.
- Support for “set-specific attributes” helps developers write concise code better focused on specific sets of cards
- Better error detection for event methods called in the wrong phase of a card action.

The first two years of the grant focused on making the Mods API more comprehensive and powerful. Year three will focus more on making the API more approachable (without sacrificing power) in order to support cultivation of a larger Wagn developer community.

5.2 RESTful Web API

Our RESTful Web API saw few changes in year two; the process of simplifying and unifying this API was largely completed in year 1.

The most significant adaptation was an upgrade intended to allow Wagn to be used as a plugin (specifically an “engine”, in ruby parlance) for other sites built using Ruby on Rails. The API was adapted such that it could be used even within the context of another application.



6 Looking Forward

The following Dev Plan(Figure 6.1) is from a live document regularly updated to insure compatibility with the priorities of the broader WikiRate Project. If you compare this document with the attached WikiRate Content Roadmap, you will notice many parallel streams, though, naturally, functionality must be developed before the community can use it.

Figure 6.1. WikiRate Development Plan

		WikiRate		Wagn/Decko			
Year	Month	Strategic emphasis	Detail	Strategic emphasis	Version	Details	
2015	October	Implement ratings!	new imports (including Industries) better filtering	Support Ratings		(all resources on ratings)	
	November		demo/test: ratings, scores, calculations - merging tags/topics			(all resources on ratings)	
	December		deploy ratings introduce transparency score	Support WikiRate Community Engagement	1.18	480px modals (mobile editing) nest editor commenting overhaul	
2016	January	Engage community around ratings	polish ratings user conversions	Multilingual Support		multilingual	
	February		gamification			multilingual	
	March		visualizations		1.19	multilingual	
	April		WikiRate tickets	Final pre-Decko 1.0 enhancements	1.20	new name patterns (base, colon, slashes) new set handling	
	May		Social Media integration SEO		1.21	API improvement: views similar to events args object	
	June	Get companies involved	person-company relationships			1.22	mods as gems wagneerable views views as nest directives
	July		company inquiries	Decko Marketing Push	Decko 0.9	rebranding Wagn-> Decko	



						new landing page for deck installs
	August	Reinvigorate Wiki Side	WikiRate multilingual			new video polish documentation tutorials
	September		wiki / rate integration metric value citations		Decko 1.0	