29th June 2022 Compiler-Research weekly meeting

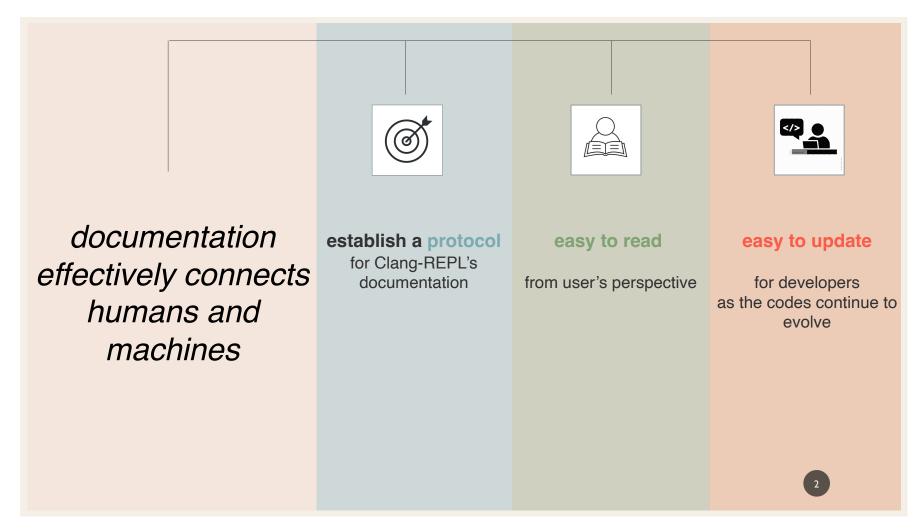
Improving the Clang-REPL documentation (June–October 2022)

Author: Sara Bellei Ph.D in Physics, Polytechnic University of Milan, Italy (2017)

Mentors: David Lange, Vassil Vasilev







Documentation effectively connects humans and machines

(https://guides.lib.berkeley.edu/how-to-write-good-documentation)

Technical Writing is a form of communication that attempts to take a technical field (like developing a new software) and convey ideas from that field as efficiently as possible, to a diversified audience, sometimes to non experts, sometimes to experts.

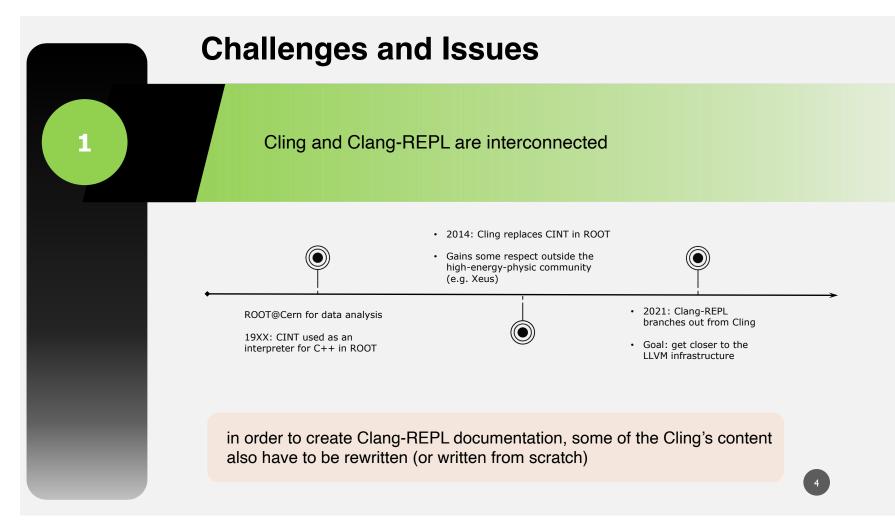
We have two goals:

- to create new content which will ensure that people who want to use Clang-REPL have the information they need to understand it, decide if it matches their needs, and operate it
- to promote in-team communication, by creating content that will allow to advance better and faster, and outgroup communication, encouraging other developers to contribute to this code

Chal		allenges and Issues	Ch Thi fac
1		Cling and Clang-REPL are interconnected	
2		who is our audience?	
3		some material already exist, but new content must be generated	
		3	

Challenges and Issues:

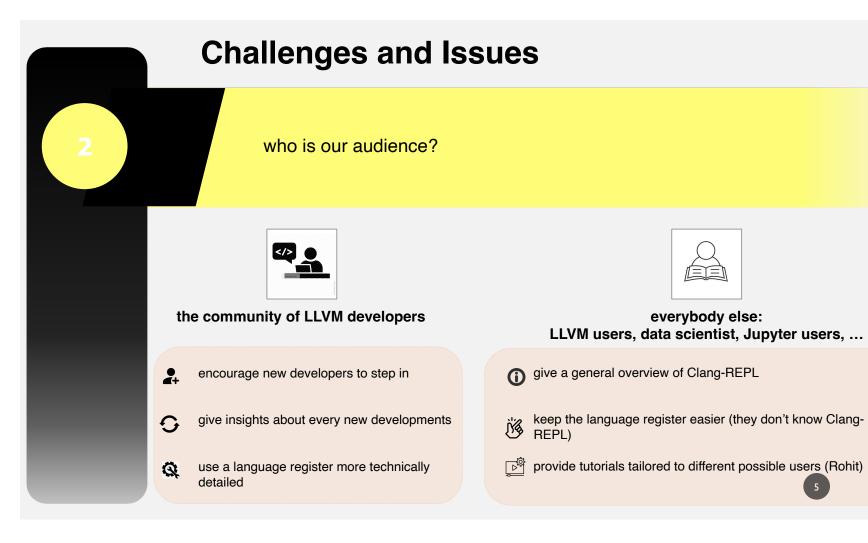
Three main challenges will have to be acced



Cling and Clang-REPL are interconnected:

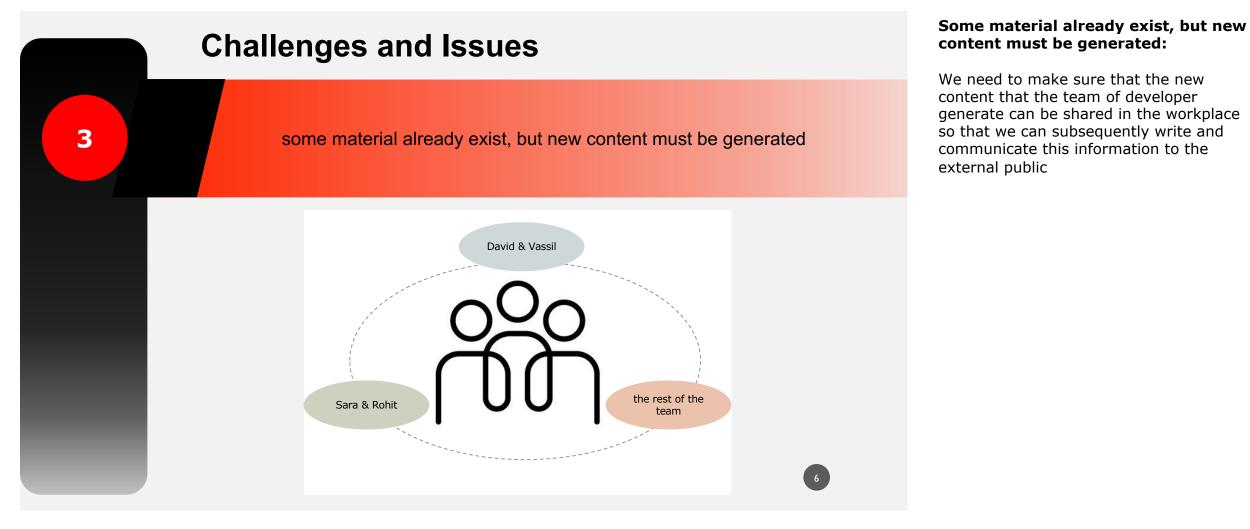
At this point, is impossible to talk about Clang-REPL without talking first about Cling.

This means that some parts of the documentation, the ones that introduce Clang-REPL should be, to some extent, written in a consistent way with the Cling documentation, because most likely a reader who wants to know what Clang REPL is will have to read something about Cling.



Who is our audience?

for both audiences we need to create guides and other forms of documentation which allows us to integrate new features into the core project as you develop them, and to describe these features in a way that makes sense to users.



Different Media Platforms

1. clang.llvm.org

2. Compiler-Researcher website

3. The LLVM Project Blog

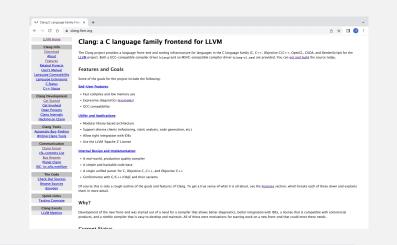
Different media platforms:

Here you can look at the media platform which will host Clang-REPL. The idea is that each of these platforms will host different content in an organized and consistent way

Different Media Platforms

1. clang.llvm.org

2. Compiler-Researcher website



- Clang-REPL is part of Clang → LLVM release instead Cling: https://rawgit.com/root-project/cling/master/www/index.html
- follows the strict LLVM developer policy

3. The LLVM Project Blog

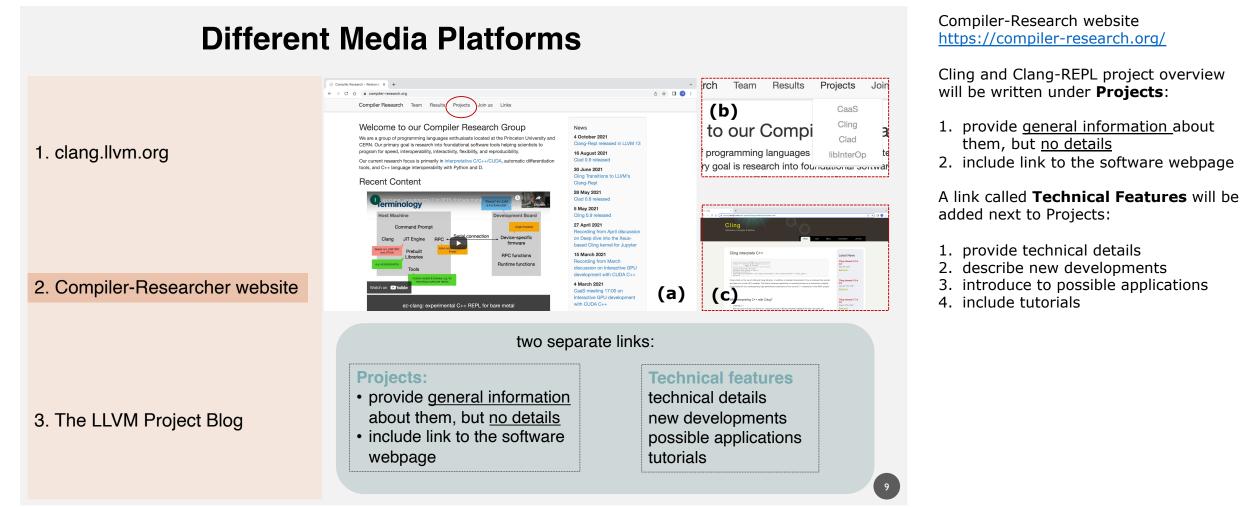
8

The Clang webpage on LLVM.org: https://clang.llvm.org/

Clang-REPL source code is found on Clang, because Clang-REPL is part of Clang, which comes as part of the larger LLVM release and not as a separate software.

LLVM has a strong policy-driven development code, which ensures that developers deliver software according to the platform's expectations.

For all these reasons, the LLVM webpage will only host the source code and it will be kept clean and readable.



Different Media Platforms

Cling Transitions to LLVM's Clin X

5

13

Vassil Vassilev

-> C & a root.cem/blog/cling-

1. clang.llvm.org

2. Compiler-Researcher website

development. acture closer to the LLVM orbit. The benefits of the solid software engineering by the LVM community have been praised widely. For example, LLVM's rigorous standards for cod Cling has emerged as a recognized capability that enables interactivity, eviews, release cycles and integration are often raised by our "external" users. We would connect two highly knowledgeable system software engineering communities - the one aroun dynamic interoperability and rapid prototyping capabilities to C++ LLVM, and the one around data analysis in HEP. The success of Cling demonstrates that developers. Cling supports the full C++ feature set including the use of mentally compiled C++ is a feature the C++ community can benefit from and the data templates, lambdas, and virtual inheritance. Cling is an interactive C++ unity needs. Finally, there are also potential synergies with projects such a blog posts → audience-focused Website \rightarrow product-focused goal: to build a community around Clang-REPL Ø how: by writing an interesting overview of the new developments then encourage the readers to follow-up by reading the official documentation

Interactive C++ with Cling - Th X +

Forum & Help Manual Blog Posts Contrib

Cling Transitions to LLVM's Clang-Repl

User the last declade we developed an interactive, interpretative U++ interpreter (Jak HerL), part of the high-energy physics (HEP) data analysis project – ROOT. We invested a significan effort to replace CINT, the C++ interpreter used until ROOTS, with a newly implemented REP based on Ihm – Cling. Cling is a core component of ROOT and has been in production since

Cling is also a standatone tool, which has a growing community outside of our field. It is recognized for enabling interactivity, dynamic interoperability and rapid pototyping capabilities for C+ developers. For example, Il you are typing C+ in a Juget notabook you are using the xeus-cling Jugete kernel. One of the major challenges is to ensure Cling's substanibility and for forst that growing community.

Cling is built on top of LLVM and Clang. Reusing this compiler infrastructure means that Cling gets easy access to new foure C++ standards, new compiler features and static analysis infrastructure. Or project organization mostly followed the LLVM community standards, but the remaining LLVM-specific customizations, while kept at minimum, are new costly for .

stainability and development. For example, it is time consuming to move to newer LLVM

ons and release Cling as following the LLVM release schedul

130 June 2020

Goals

→ C O B blog.llvm.org/posts/2020-11-30-

The LLVM Project Blog

LLVM Project News and Details from the Trenches

By Vassil Vassilev Nov 30, 2020 <u>#Cling</u> 11 minute read

Interactive C++ with Cling

Interactive C++ with Cling

The C++ programming language is used for many numerically intensive scientific applications. A combination of performance and solid backward

inconsistent with rapid application development. Exploration and

prototyping is slowed down by the long edit-compile-run cycles during

compatibility has led to its use for many research software codes over the past 20 years. Despite its power. C++ is often seen as difficult to learn and The LLVM project blog https://blog.llvm.org/

0 🗙 🛛 🥥

0040

About Posts Tags Ilvm.org

We will also keep posting on blogs, in particular on the LLVM blog.

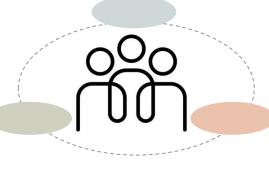
A technical blog is the best form of advertising for the software, because unlike the documents describing technical features in detail, it will include plenty of keywords, visually appealing pictures, infographics and have a writing style that is more casual, as if the writer is talking to you. Their priority here is to reach a wider audience.

It will be used for describing Clang-REPL applications, and to keep the LLVM community updated regarding new Clang-REPL's features, and we will describe how these development will positively impact on the Clang project

3. The LLVM Project Blog

Conclusion:

it is often only in the process of writing that certain critical aspects of the work become apparent



some questions I ask you to think about:

what is my contribution to Clang-REPL?

have I achieved the key result I want to build my work around?

how do I achieve the key result

how will the project benefit from my contribution?

is there a possible application for it?

from the Clang webpage:

(...)evidence of a significant user community should be provided: This is based on a number of factors, including an existing user community, the perceived likelihood that users would adopt such a feature if it were available, and any secondary effects that come from, e.g., a library adopting the feature and providing benefits to its users

Conclusion:

This presentation shows opportunities to improve the working relationship between the development teams and the technical writing teams.

From a developer point of view, having a consistent and clear written documentation about your software will force you to get into the habit of putting abstract concepts into understandable words.

This improves the way you conceptualize and deal with technical problems, and you communicate those problems to others.

For all these reasons, technical writing should be considered as a key factor to contribute to the overall development and expansion of the Clang-REPL project.

Thank you for your attention!