Master Thesis Kick-off Presentation

“Improving the Software Architecture Documentation Process for MediaWiki software”

23.03.2015, Ankitaa Bhowmick

Software Engineering für betriebliche Informationssysteme (sebis)
Fakultät für Informatik
Technische Universität München

wwwmatthes.in.tum.de

Administrative Matters

Time:

February15th, 2015 to August 15th, 2015

Supervisor:

Prof. Dr. Florian Matthes

Advisor:

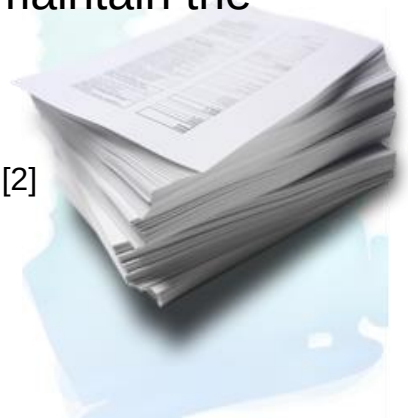
Klym Shumaiev

Agenda

1. Motivation
2. Research Questions
3. Methodology
4. Evaluation
5. Timeline
6. Reference

Motivation

- A good software architecture is the focal point of a good software
- “To make this software maintainable and extendable, a state-of-the-art SA documentation process is required.” - The Software Sustainability Institute
- “SA documentation needs to satisfy the needs of different stakeholders” - Documenting Software Architectures: Views and Beyond
- SA document needs to be up-to-date with changes
- “Software complexity grows with time and it becomes tedious to maintain the documentation” [1]
- Lack of documentation is the downside of open source Software [2]



Motivation

Current state-of-the-art :

- MediaWiki is a complex evolving OSS in recent times
- SA documentation of MediaWiki SA is available on “mediwiki.org” [3]

Available features:

- Commenting / Discussion (Talk)/ History
- Editing - CRUD features
- Access rights and permissions – based on “User Groups” [12]
- Responsibilities - Developers as Maintainers [4] , Gerrit reviewers

Motivation

sebis

Drawbacks of the current DSA process

(derived from interview and email communications with MediaWiki architecture committee members)

- Outdated parts / missing parts of documentation

- Limitations of open source software
  - Irresponsibility to maintain documentation
  - Informal process (Phabricator, IRC channel)

Goal

An improved process to create mid-level SA documentation on Mediawiki.org that automates the tracking and maintenance of documents

Research Questions

RQ1 : How software architecture documentation process can be improved for Wikimedia Software?

RQ2 : What state-of-the-art architecture documentation process (methodology, tools) are available in the industry that meet domain-specific requirements – e.g. Open Source S/W ?

RQ3 : What are the quality characteristics and metrics for evaluation of the software architecture documentation process?

RQ 4 : Which specific requirements of Wikimedia stakeholders should be met by documentation process for Mediawiki SAD ?

RQ 5 : What process can be followed to automate the quality assurance of SA documentation in OSS

Methodology

[Design Science Research in Information Systems]

Methodology

- Find stakeholders
- Interview stakeholders
- Derive new requirements

[Design Science Research in Information Systems]

Methodology

- Survey the available state-of-the-art
- Define metrics for evaluation
- Identify the current system's properties

[Design Science Research in Information Systems]

Methodology - Requirements

Requirements derived for SA documentation process

Maintainability – The process should allow code owners to CRUD the SA documentation.

Evolution and Process Maturity – The documentation process can handle the software architecture growth.

Tracking - An automated control to detect module-level changes in the code and notify code/module owners.

Access rights, roles – Maintain a role-based access to SA document corresponding to the code/module owners.

Accessibility – Provide features like index, glossary, etc. for easy navigation and search of related elements/modules/packages

Versioning and Consistency – Automated consistency resolution to maintain only one up-to-date version of the SA document.

Methodology

- Design an effective solution as per requirements
- Improve the documentation process

✓ Track changes at code level

✓ Send notification to Update/ Add documentation

✓ Analyze the history logs of changes made to the content

✓ Create a dashboard on mediawiki.org

[Design Science Research in Information Systems]

State of the art

Documentation process of MediaWiki Software

Developer

Code commits

GitHub

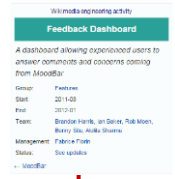
Automatic code documentation generation

Doxygen

Questions

Mails

Comments



#irc

Task creation/update

PHABRICATOR

MediaWiki-Documentation project

CRUD operations

MediaWiki
Because ideas want to be free.

History comparison

Talks

Review (?)

Next iteration

State of the art

Documentation process of MediaWiki Software

Developer

Code commits

GitHub

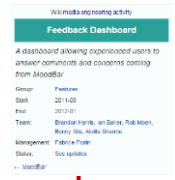
Automatic code documentation generation

Doxygen

Questions

Mails

Comments



#irc

Task creation/update

PHABRICATOR

MediaWiki-Documentation project

CRUD operations

MediaWiki
Because ideas want to be free.

History comparison

Talks

Review (?)

Next iteration

State of the art

Documentation process of MediaWiki Software

Developer

Code commits

GitHub

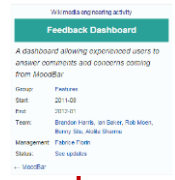
Automatic code documentation generation

Doxygen

Questions

Mails

Comments



#irc

Task creation/update

PHABRICATOR

MediaWiki-Documentation project

CRUD operations

MediaWiki
Because ideas want to be free.

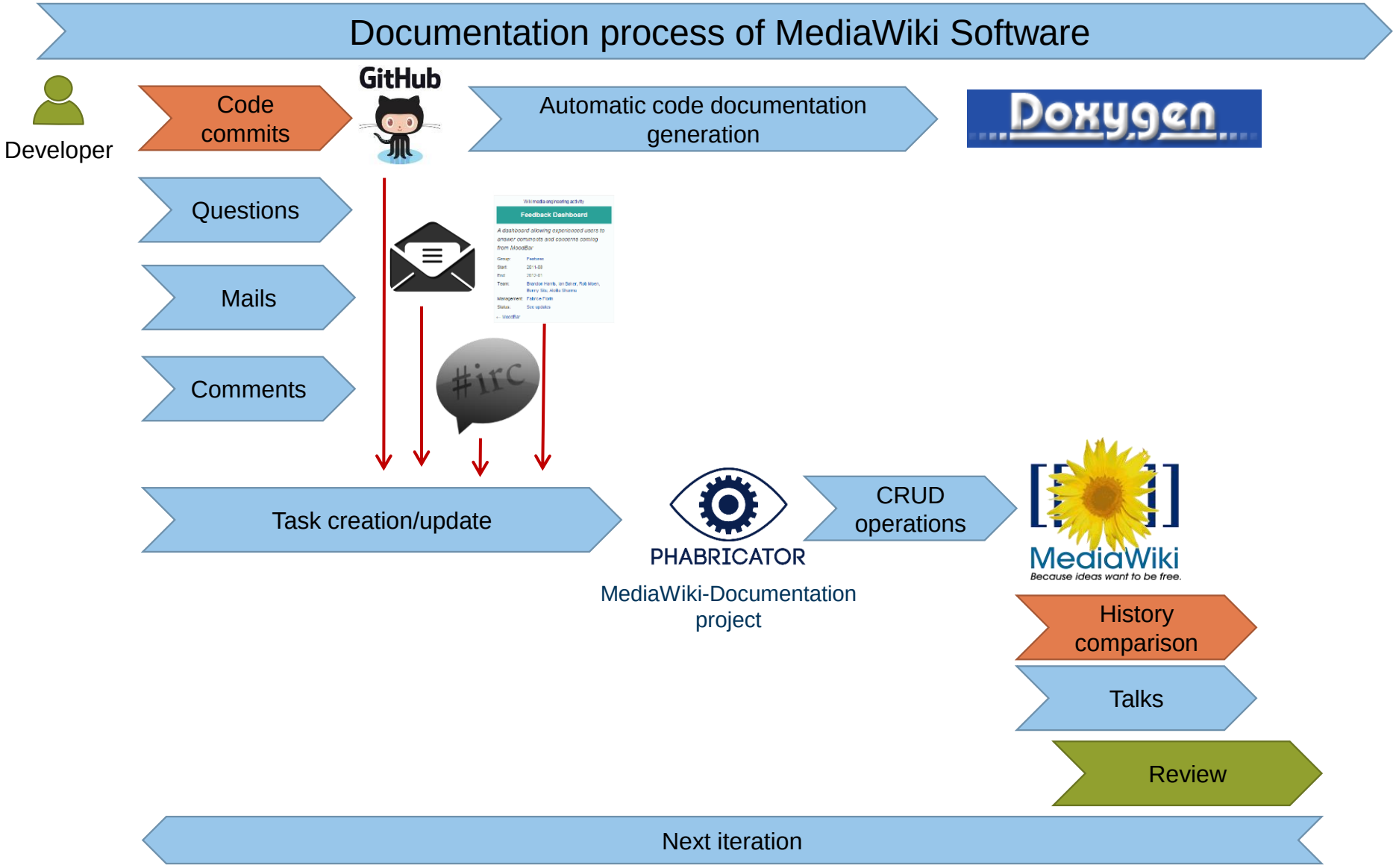
History comparison

Talks

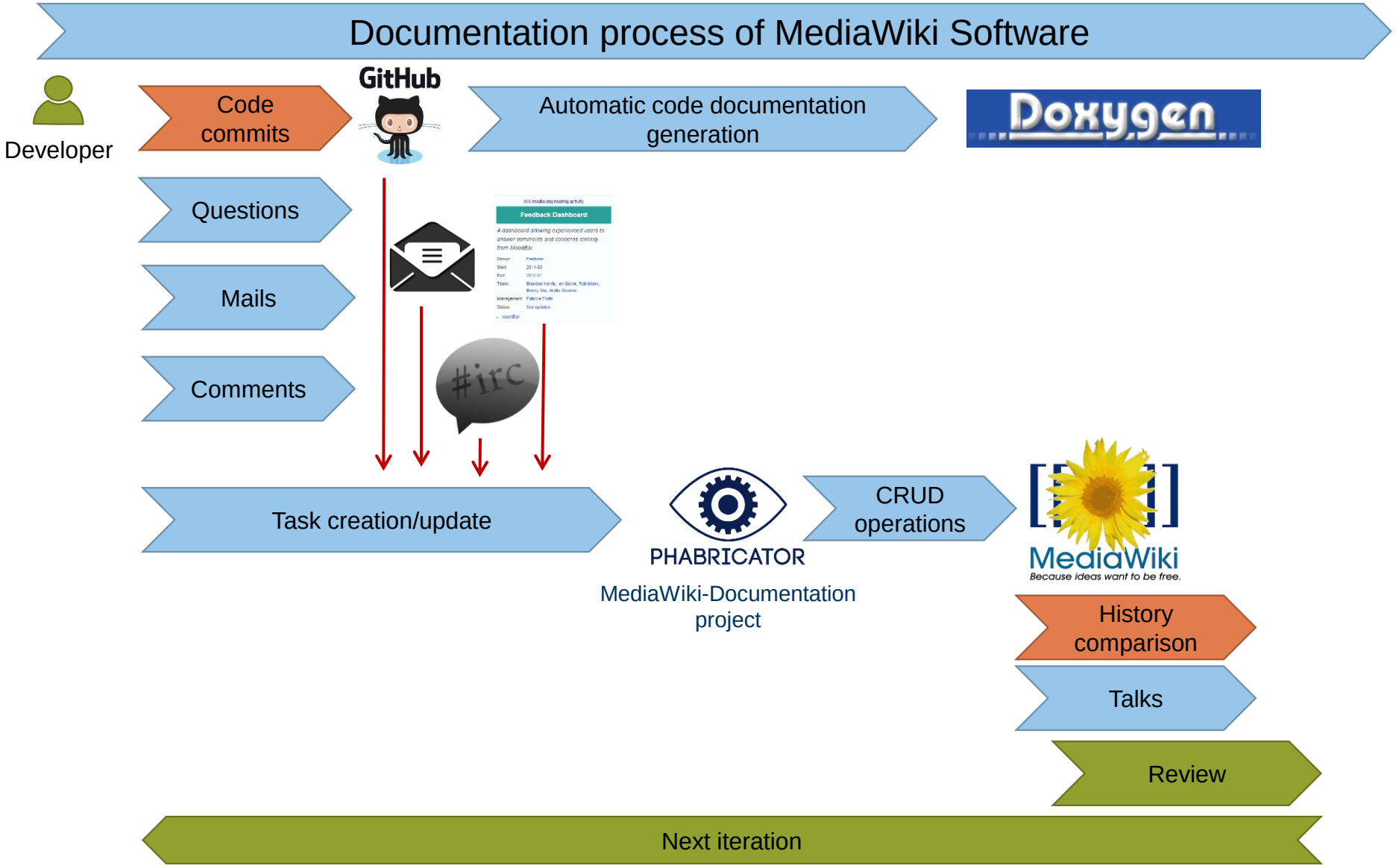
Review (?)

Next iteration

State of the art



State of the art



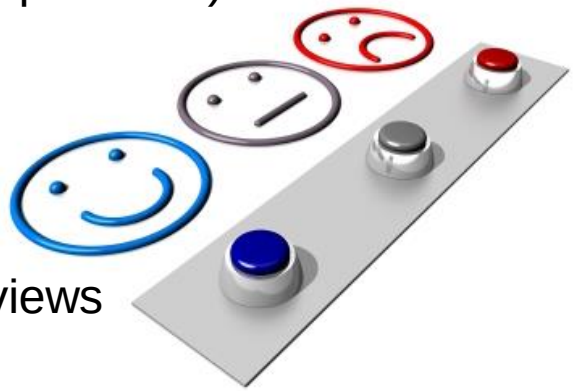
Methodology

- Integrate with the existing system and test.
- Conduct surveys
- Evaluate the improved process

[Design Science Research in Information Systems]

Evaluation

- Evaluate requirement satisfaction (answers to research questions)
  - Is the documentation process improved?
  - Does the process satisfy the expectation of OSS?



- Evaluate stakeholder satisfaction with Survey and Interviews
  - Quality assurance in the improved process - feedback of MediaWiki architecture committee
  - Process stability validation - feedback from MediaWiki developers
  - Ease of use / understandability of the documentation process - new developers
  - Acceptance in the wiki community – interviewing people from different roles and responsibilities (internal employees and external developers)

Metrics for process Evaluation

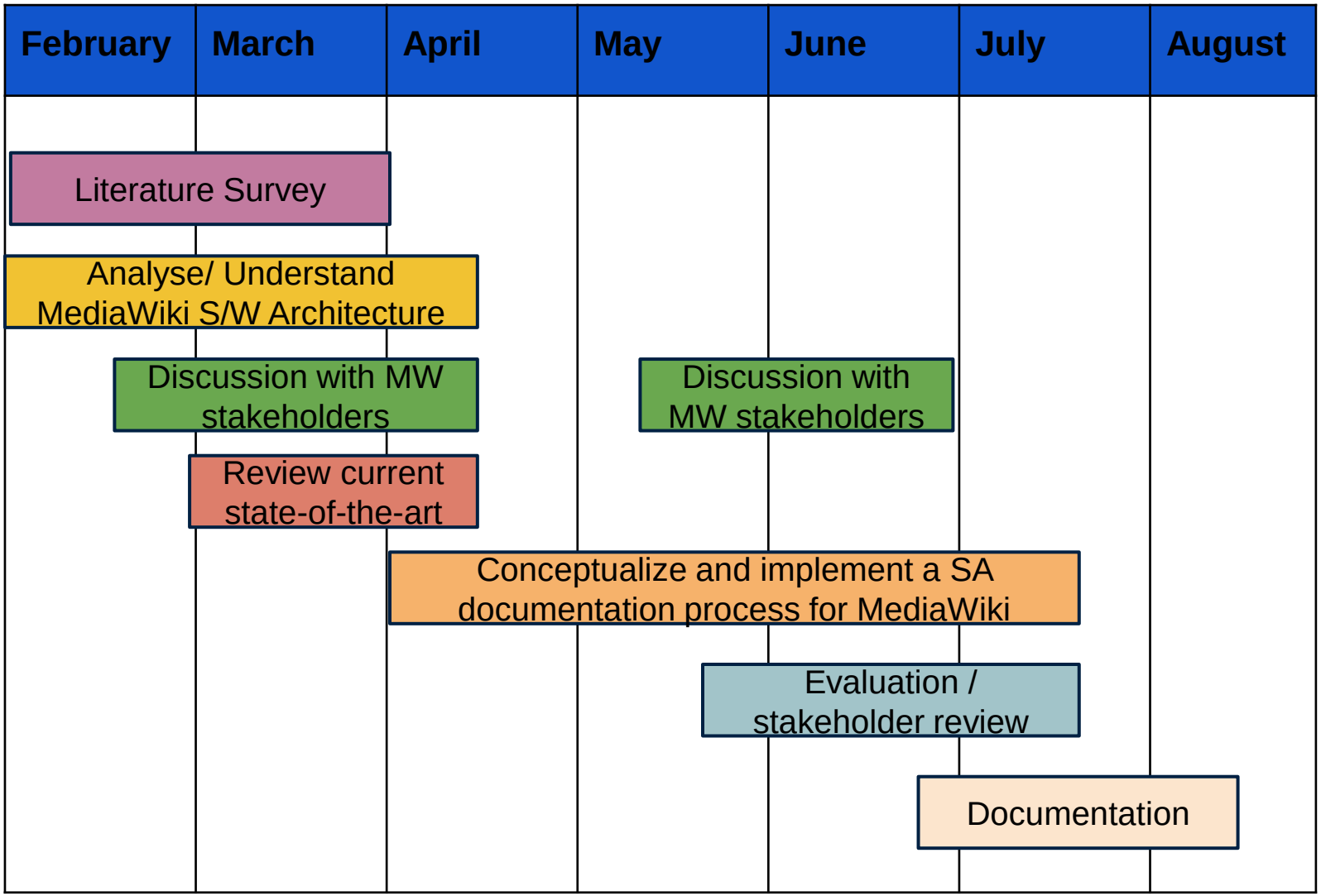
- Maintenance efforts and costs vs. capacity[11]. For example:
  - Personnel
  - Time
- Is the process adequate for its intended purpose?
- QualOSS standards for process evaluation[5]. For example:
  - Review process
- Process features also need to be evaluated[9]. For example:
  - Architecture tracking
  - Multiple user support
  - Capture and reason

Methodology

- Get feedback on the improvements
- Discuss stakeholder satisfaction

[Design Science Research in Information Systems]

Timeline


February
March
April
May
June
July
August
Literature Survey
Analyse/ Understand MediaWiki S/W Architecture
Discussion with MW stakeholders
Discussion with MW stakeholders
Review current state-of-the-art
Conceptualize and implement a SA documentation process for MediaWiki
Evaluation / stakeholder review
Documentation

References

1. http://oss-watch.ac.uk/resources/archived/documentation
2. Fundamental issues with open source software development - Michelle Levesque
3. http://www.mediawiki.org/wiki/Manual:MediaWiki_architecture
4. https://www.mediawiki.org/wiki/Developers/Maintainers
5. The QualOSS Open Source Assessment Model Measuring the Performance of Open Source Communities - Martín Soto and Marcus Ciolkowski
6. A comparative study of architecture knowledge management tools - Babar et.al
7. Documenting Software Architecture - Views and Beyond
8. How Do Open Source Communities Document Software Architecture: An Exploratory Survey – Ding et.al
9. Software Process: A Roadmap - Alfonso Fuggeffa
10. Design Science Research in Information Systems
11. A systematic review of software architecture visualization techniques – Babar et.al
12. http://www.mediawiki.org/wiki/Manual:User_rights#List_of_Groups

Thank you for your attention!
Questions?