

DOI: https://doi.org/10.48009/4_iis_2021_117-133

DevOps paradigm -a pedagogical approach to manage and implement IT project

Abhijit Sen, *Kwantlen Polytechnic University, abhijit.sen@kpu.ca*

Laura Baumgartner, *OTH Regensburg, laura.baumgartner@st.oth-regensburg.de*

Katharina Heiß, *OTH Regensburg, katharina.heiss@st.oth-regensburg.de*

Cornelia Wagner, *OTH Regensburg, cornelia1.wagner@st.oth-regensburg.de*

Abstract

DevOps, the widely used term in software industry, integrates the Development and IT Operations activities to frequently deliver, deploy, and release quality software features. DevOps approach emphasizes collaboration among Developments and IT operations teams throughout System Development Life Cycle (SDLC). The DevOps process is supported by wide variety of tool chains for various phases of SDLC. There exist many DevOps models. However, in this paper authors use a simple four phase pedagogical models to demonstrate principles of DevOps. In this paper authors attempt to show how DevOps principles can effectively be used to manage and implement business problems in classroom setting. Specifically, DevOps methodology is applied to manage develop and implement a small web application. This pedagogical approach is specially aimed at students who do not have prior experiences and skillsets in applying DevOps methodology and associated toolsets to every stages of SDLC. At the conclusion of the project, students gained valuable insights on how to apply DevOps principles to business problems and to select and use commonly used state of the arts tools to plan, manage, build, test, monitor, deploy tasks at every stages of DevOps. The authors also discuss the limitations and practical issues related to implementing DevOps within classroom settings.

Keywords: Continuous delivery, Continuous Deployment, Continuous integration, Continuous Monitoring, DevOps, SDLC, KPIs

Introduction

In the age of digitalization, more and more companies are turning away from the classic waterfall work model of software development to more powerful and modern agile models. One of these models is called DevOps lifecycle. This represents different phases of a project and links them into a cycle. There are a number of software tools that support this cycle and make it easier to work together on a project. However, students in traditional Computer Science or IT programs in Universities hardly get opportunities to study the complete DevOps principles and apply and select tools that can be used throughout different DevOps stages. To meet the demands of the industries, and give students opportunities to learn and practice various DevOps tools, the course DevOps Principles and Practices was offered at the OTH Regensburg, Germany. The course includes a required technical term project of suitable scope (approx. 3-4 weeks' effort) that will cover all phases of the DevOps lifecycle and be supported by appropriate industry standard tool sets. The project intends to simulate DevOps environment in class room setting. The choice of tools to be used is left to the project teams themselves. In this paper, one of the student's project is discussed.

There are number of software tools available in the market that support various DevOps functionalities. Tool chains provide an integrated set of tools to build, deploy and manage the applications. Students can create tool chains that include customized services, open source tools, and third party tools that make

development and operations repeatable and easier to manage. The purpose of this project is to select a set of tools that will support and automate all activities of DevOps Life cycle.

The objective of this paper is to show how students can learn DevOps skillsets in a classroom environment to manage and implement a small web development project. Students applied commonly used industry standard DevOps tools to manage and develop the project using the principles of DevOps paradigms.

Literature Survey

DevOps paradigm integrates software development (Dev) and information technology operations (Ops) using sets of principles and practices to deliver high quality software product with rapid frequencies. Davies, Daniels (2016) describe the details of DevOps principles, their implementation and approaches to choose various tools to facilitate DevOps workflows. Many different models of DevOps life cycles can be used in practice. Many of them are similar and only differ slightly. One such model was described in Bluestone & Drew, Sen (2021) that uses eight stages. However, for this project we have utilized the model used in Sharma & Coyne (2015) that uses four stages in DevOps life cycle as shown in Figure 1.

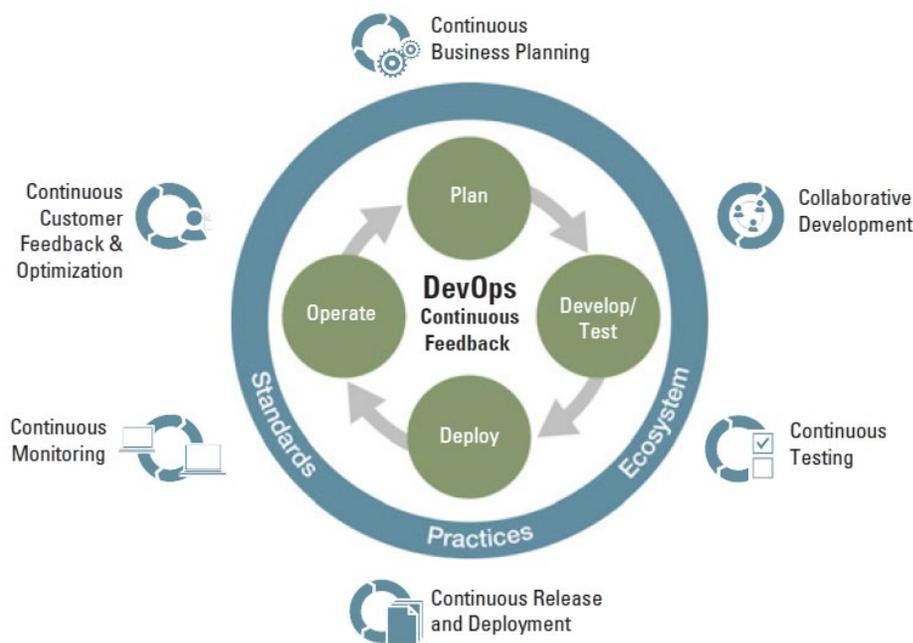


Figure 1. DevOps Life Cycle (Sharma, Coyne).

As shown in Figure 1, DevOps consists of four stages- Plan, Develop & Test, Deploy, and Operate, where DEV consists of {Plan, Develop & Test, and Deploy} segments, and Ops consist of {Operate} segments. All four phases offer opportunities for continuous improvement and, for example, to better consider customer feedback in the long term. However, there are many challenges one may face in implementing DevOps as described in Leite et.al (2019) particularly in selection of tools from the vast arrays of tools that are available for various DevOps stages.

In this paper attempt is made to integrate DevOps principles in our example application using some popular tool sets. Some commonly used tools for successful DevOps implementation of DevOps are described in Armlin (2021), Dennis (2020), and Sen (2021). To keep the paper concise and to emphasize salient features of DevOps pedagogy, many details of the implementation are purposely left out in this paper.

The DevOps-Lifecycle starts with planning. In the **Planning** phase, the focus is on setting business goals and modifying them on the basis of customer feedback. The **Develop/Test** phase includes two major practices: Collaborative Development and Continuous Testing, which forms the core of the development and quality assurance abilities. In the **Deploy** phase, continuous integration will be taken one step further. By setting up a development pipeline, also known as a delivery pipeline, continuous deployment can be carried out and new functions can be quickly released to customers (Continuous Release and Deployment).

The **Operate** phase comprises two important practices: Continuous Monitoring and Continuous Customer Feedback and Optimization. Continuous Monitoring provides data and metrics for all stakeholders of the software product, including operational staff, such as quality assurance, development and business departments. The monitoring can be executed throughout the entire DevOps cycle, and provides information and guidance on where changes or improvements are needed to deliver a good product. Among the most important information in software development is information about how a customer uses the product and what feedback the customer gives. Only when this information is available appropriate and concrete measures can be taken in the development of the product

The rest of the paper is organized as per the following manners.

The actual Project developed is discussed in section “Project Description”. The hardware, operating system and browsers details used in the project are described in section “Project Environment”. Section titled “Tools Used for DevOps Project” discusses selected tools used for application development and deployment. The output of important DevOps stages is shown in section titled “Result” followed by conclusion in section titled “Conclusions”. All the Figures are provided in the **Appendix** for ease of referring.

Project Description

The team consists of three team members of Business Informatics program. For the project they mime the imaginary company *Creative Brains*³, which wants to create a simple website using the DevOps-Lifecycle. The project aims to demonstrate how efficient, flexible and transparent collaboration can be by using the DevOps principles and practices.

Team Member 1 takes on the role of CEO. Team Member 1 has practical experience in working with Atlassian Software as a student trainee. Team Member 2 is a trained web designer and brings know-how in the area of website creation. Team Member 3 did get to know DevOps during internship as Scrum Master, bringing some experience from the agile world into the team. The individual stages of the lifecycle are divided into the strengths of the individual team members. The software is still tested together to achieve the greatest possible learning factor for all participants.

Table 1. Project team member Responsibilities.

Team Member 1	Plan, Operate
Team Member 2	Develop, Test
Team Member 3	Deploy

The completion of a website is the main goal of the project, but in reality, only a by-product. Proof of this is also the use of a ready-made, open source template (<https://html5up.net/dimension>). On the website there will be a subpage for each team member as well as the pages required in Germany for imprint and data protection. Further project milestones will be highlighted in the next sections of the paper and documented by screenshots.

Project Environment

The project is carried out on a total of three windows computers. Due to corona virus pandemic there were no personal meetings. All communication was done via Zoom meetings and WhatsApp group chat in the beginning - with project start also increasingly supported by using Atlassian products (especially Confluence and Jira see – Table 2, Table 3 below). The basic configuration of the project environment is as follows:

- Hardware devices (Windows-based):
ASUS UX360UAK, Windows Surface Pro 7, and HP ProBook 450 G4 computers
- Mobile Operating Systems: iOS 13.5.1. and Android 9
- Browser:
 - Main browsers: Google Chrome (Version 83.0.4103.116, Official Build, 64-bit), Firefox (Version 78.0.2, Official Build, 64-bit).
 - Additional quick tests with the following browser:
 - Microsoft Edge (Version 83.0.4103.116, Official Build, 64-bit).

Tools Used for DevOps Phases

Different well established industry standard tools are used for implementing the various stages of DevOps life cycles. Table 2 list the tools used for different DevOps stages.

Table 2. Selected Tools Used at various DevOps stages

Name	Website	Version	DevOps Stage
Atlassian Confluence	www.atlassian.com/de/software/confluence	Web Free Cloud Version, Mobile App Version for Cloud 1.78.11 (Android), 109.0.1 (iOS)	Plan (Planning, Collaboration), Operate (Customer Feedback, Optimization)
Atlassian Jira	www.atlassian.com/de/software/jira	Web Free Cloud Version - Mobile App Version for Cloud 53.0.436	Plan (Planning, Collaboration), Operate (Monitoring, Optimization)
Bitbucket	bitbucket.org	Bitbucket 7.4	Development
Markup Validation Service	https://validator.w3.org/	W3C Markup Validator, v1.3+hg.	Testing
Jenkins	www.jenkins.io	Jenkins 2.444	Continuous Release and Deployment
Google Analytics	analytics.google.com	-	Operate (Continuous Monitoring, Customer Feedback)

For details of these tools one can refer to appropriate websites specified in these tables.

For comprehensive lists of DevOps tools one may refer Sen (2021), and Periodic Table of DevOps Tools in <https://digital.ai/periodic-table-of-devops-tools>.

Results

Continuous Business Planning with Confluence and Jira:

Confluence is a web application for collaboration and provides a variety of templates. Confluence is used to create conceptual planning page which three team members can access (Figure 1 and 2). Figure 3 shows sample road map of the project outlining various DevOps stages with appropriate completion dates.

Furthermore, team members will always be notified when someone edits pages (Figure 4). The administrator of the confluence area can also set access rights of individual team members to ensure confidential and internal information can be protected against unauthorized access (see Figure 5).

Jira from Atlassian is used in the project for issues tracking (Figure 6), and can be easily integrated with Confluence. In addition, *tasks* can also be created to which many individual issues can be added and linked. Versions of products can also be displayed in a structured manner and, for example, structured according to release as in our project (Figure 7). Jira is also available as a mobile version (see Figure 8 – 10). In order to keep track of the current work in progress, the app complements the web application with continuous accessibility. As soon as an item is available, and tested, it is ready to be deployed, as explained later. The product owner can also view which functions are included in which release versions directly in Jira (Figure 11-12).

Collaborative Development:

For collaborative development the web application Bitbucket from the Atlassian was used to perform all version control functions. Bitbucket is compatible with the other two web applications. The three applications together form a well spun web of information that helps the development team to work collaboratively. Figure 13 shows how using *Bash* command one can store HTML template to the Bitbucket repository.

The collaborative work is supported by *Branches*. The *master* branch is always protected and contains the original files. Figure 14 shows a *prod* branch is created, which will contain the files to be edited, which are committed and merged. The *master* branch is always protected and contains the original files. By linking to Jira Figure 15 shows an example of the current state of Bitbucket dashboard, open work packages and pull requests. By integrating with Jira, everyone can keep track of the project and open tasks.

Continuous Testing:

To test the generated and merged code, the W3C Validator was used. Figure 16 shows example of testing code by just copying the code and paste it into *Direct Input*. The service tests code for syntax errors and marks them for potential errors. Additionally, the web application also warns of possible difficulties with the code (Figure 16-18). The warnings can be created in Jira as a work package (see Figure 19) in a sprint and one can be sure that this possible problem is still being worked on. The whole team now also knows that there is still something to be done. The warnings can be created in Jira as a work package (Figure 19) in a sprint to alert team members that there is still something to be done.

Continuous Release and Deployment:

Continuous Delivery represents the automatic release to the repository and then it is automatically deployed to production environment with the aid of Continuous Deployment. To support continuous release of small pieces of work, Jenkins was used. Atlassian Bitbucket can be integrated easily with Jenkins in this phase of the DevOps lifecycle.

To automate the procedure to deploy to production a job needs to be created. This job (Figure 20 and 21) is connected with the repository in Bitbucket. Furthermore, every build is triggered if something is pushed to this repository. Figure 22 shows build job run successfully. Using the automatic process, developers are relieved from rudimentary tasks and can concentrate on other important issues.

Continuous Monitoring:

Monitoring tools are used to keep track of all processes, progresses and weaknesses. This can be interpreted in different ways. On the one hand, it is essential for the later consideration of customer feedback to find out how the customer uses his own product, i.e. our website. For this reason, Google Analytics is used below to analyse and monitor the use of the website - in real time.

In addition, it is also useful to look beyond the obvious, i.e. the product. Monitoring your own work can also be useful and lead to important insights. For this reason, we have also looked at the possibility of Jira Report.

Google Analytics:

Google Analytics offers a variety of possibilities for websites to be analysed. Since a website was created as a product in our project, Google Analytics was the obvious choice for monitoring about the satisfaction of visitors with the clarity and structure of the website. In Figure 23 in the start page some important KPIs such as the number of users, sessions, bounce rate and session duration are presented.

Jira:

In Jira, there is a large number of reports that can be generated (see Figure 24), which are primarily intended for the management level, but also for a Scrum Master, in order to find out about the changes in the development process that could lead to improvements.

Both monitoring the website and monitoring one's own work can provide interesting and important insights. For the internal monitoring of the team's work, Jira is a good choice - if it is already in use in planning. Similarly, the insights gained through Jira reports can be further processed. For a team of three, these findings are not very meaningful in the example project, but in larger teams working on projects over a longer period of time, this is certainly a very good opportunity to improve further with the findings and optimize processes based on them.

Continuous Customer Feedback and Optimization:

Furthermore, it is essential to obtain customer feedback and consequently optimize the product. With DevOps, customer feedback can always be obtained promptly and at best after each sprint. Adjustments can then be initiated to meet the actual needs of the customer. Customer's feedback could be obtained

through meetings that can be held with suitable communication tools (Zoom, Slack, MS Teams etc.) or best in the real test environment of the customer. The feedback must be recorded. Additional indirect feedback through the use of monitoring tools, such as Google Analytics, reveal further findings that need to be considered in the further course of the project.

The insights gained should in turn be recorded in a sustainable manner. To this end, not only the above-mentioned reports should be documented as feedback, but also meetings and direct agreements with customers. This can again be easily achieved with Confluence and therefore requires no further tool and no further costs (see Figure 25).

The results of the analysis and the documentation of direct customer feedback can then be collected and structured centrally in Confluence.

Discussions & Conclusions

The goal of this project was to get to know DevOps principles and how these principles can be implemented in practice. The DevOps model used consists of the steps: Continuous Business Planning, Collaborative Development, Continuous Testing, Continuous Release and Deployment, Continuous Monitoring and Continuous Customer Feedback and Optimization. This particular model is selected because of its simplicity and ease of use. But DevOps facilitates work to be done iteratively, so after every sprint it is possible to append new issues. With the clearly defined process and availability of tools, collaboration with other team members was easy in spite of the fact that the team had to work remotely. Communication with team members were never an issue.

The students have limited exposures to DevOps tools which posed problems in selecting appropriate tools. The appropriate selection of tool chain, specifically the use of several products from one vendor (Atlassian), for different stages of DevOps life cycle helped to reduce the problem of integrating the tools. Atlassian has good compatibility with other tools from different vendors. Using Confluence collaborative platform from Atlassian helped team in capturing project requirements, assigning tasks to specific team members, and manage several calendars at once. Jira from Atlassian is a flexible issue tracking tool that helped teams plan, manage and report on their work. Confluence and Jira complement each other, enabling Confluence users the ability to view, interact with, and reference Jira issues from a Confluence page. Combining these tools with Bitbucket increases the transparency as every change needs to be approved by other team members. Jenkins is a widespread deployment tool supporting CI/CD. Jenkins is easy to use for beginners as there is no need of writing all steps on your own for the pipeline. With Jenkins it is easy to configure jobs which can be put together to one pipeline. E-Mail-Notification and monitoring features improve the deployment. To monitor the website continuously Google Analytics and Jira-were used that helped improve the work in a team. Google Analytics provides insights into the use of the website. With the help of KPIs meaningful results can be drawn about the use of the website by the user. Another important thing is to gain feedback for the next sprints. As Google Analytics, Jira, and Confluence offer possibilities to get feedback no other tools were therefor implemented. Jira can be seen as inside feedback, Google Analytics as customer feedback and in Confluence all aspects can be collected and secured for the future to be considered in the next sprint. As these tools are already providing enough insights, measures can be taken to optimize. Students gained valuable insights about DevOps principles and applied state of the art tools to manage and implement the project in an academic environment.

References

Armlin, D. (2020). 9 Essential DevOps Tools for 2021, Retrieved July 9, 2021 from <https://tinyurl.com/e2xy9x42>

Atlassian (2020a). Bitbucket, Retrieved July 9 , 2021 from <https://bitbucket.org/>

Atlassian (2020b). Confluence Retrieved July 9, 2021 from <https://www.atlassian.com/de/software/confluence>

Atlassian (2020c). Jira. Retrieved July 9, 2021 from <https://www.atlassian.com/de/software/jira>

Bluestone, D, Drew, G. Electrify Web Developments with DevOps, Retrieved July 9, 2021 from <https://www.cyber-duck.co.uk/insights/resources/electrify-web-development-with-devops>

CD Foundation (2020). Jenkins, Retrieved July 9, 2021 from <https://www.jenkins.io/>

Davies, J, Daniels, K (2016). Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, O'Reilly Media; 1st edition

Dennis, D. (2020). DevOps Tools Every Engineer Should Know, Retrieved July 9, 2021 <https://dev.to/daviddennis02/devops-tools-every-engineer-should-know-hf2>

Google Analytics (2020). Google Analytics, Retrieved July 9, 2021 from <https://analytics.google.com/>

Lette, L, Rocha, C, Fabio, K, Dejan, M, Paulo, M (2019), A Survey of DevOps Concepts and Challenges, Retrieved July 9, 2021 from <https://arxiv.org/pdf/1909.05409.pdf>

Sen, A. (2021). DevOps, DevSecOps, AIOps- Paradigms to IT Operations. Evolving Technologies for Computing, Communication and Smart World, Pages 211-221, Springer

Sharma, S. & Coyne, B. (2015). DevOps for Dummies. Hoboken, NJ: John Wiley & Sons, Inc.

W3C (2020). W3C Validator, Retrieved July 9, 2021 from <https://validator.w3.org/>

Appendix

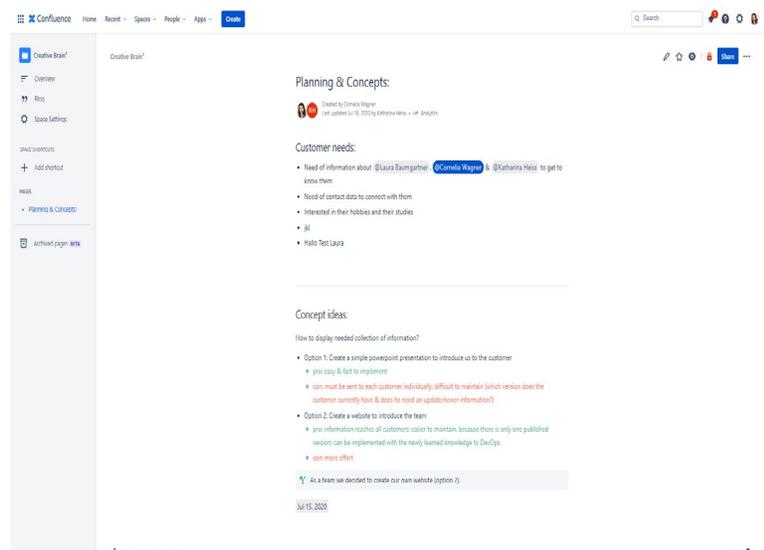


Figure 1. Created page to collect ideas and create plans in a collaborative team.



Figure 2. Mobile view within the Confluence App (on an Android device).

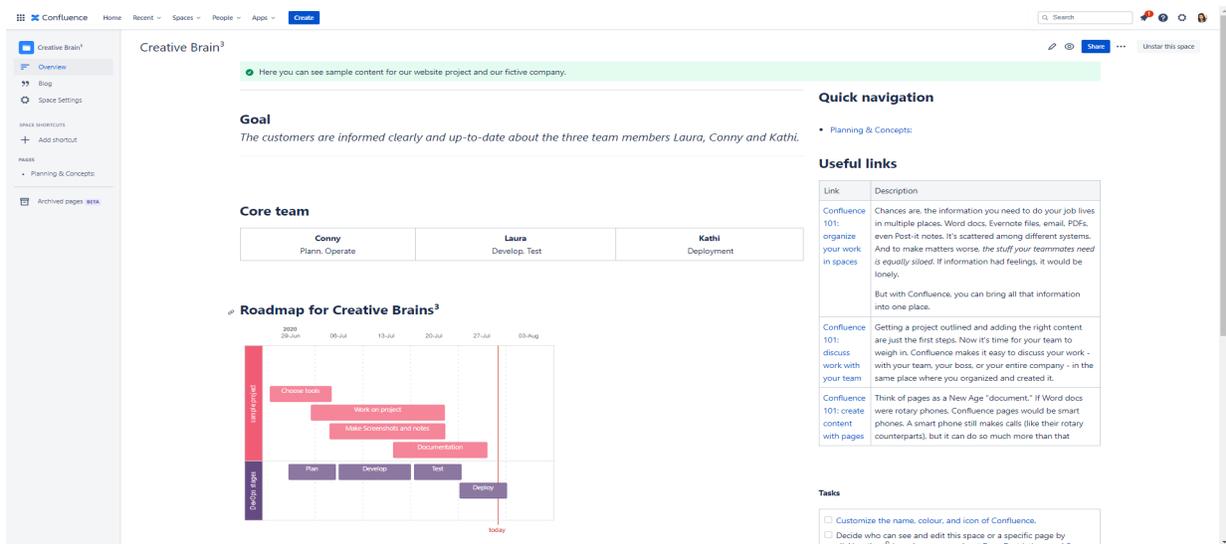


Figure 3. Roadmap of the project created using Confluence template.

Issues in Information Systems

Volume 22, Issue 4, pp.117-133, 2021

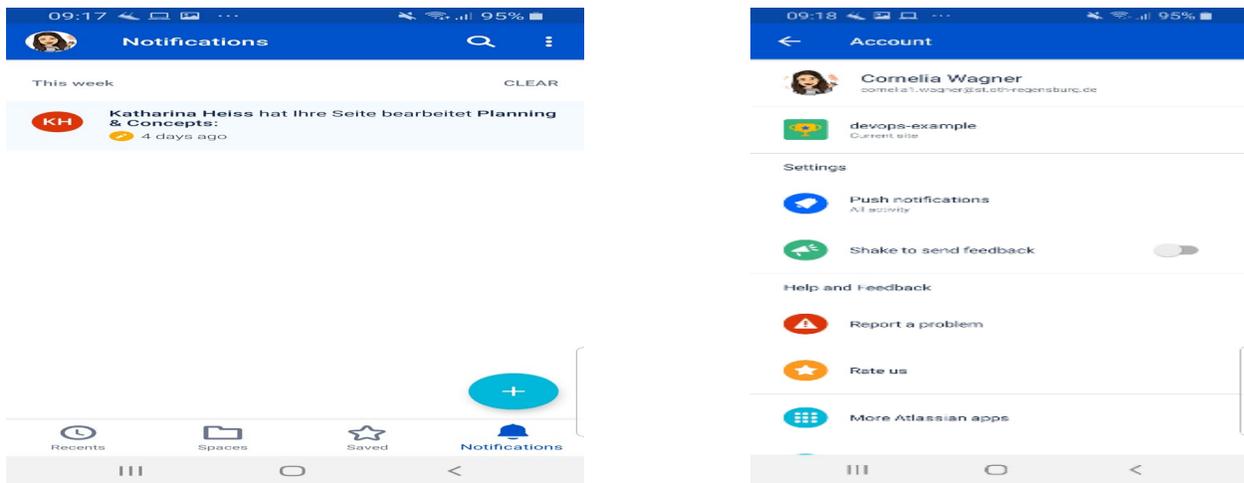


Figure 4. Notification and settings for further messages.

User	Last activity	Status	Actions
 Cornelia Wagner ORG & SITE ADMIN cornelia.wagner@st.oth-regensburg.de	July 20, 2020	Has site access	Show details ⋮
 Katharina Heiss TRUSTED katharina.heiss@st.oth-regensburg.de	July 16, 2020	Has site access	Show details ⋮
 Laura Baumgartner TRUSTED laura.baumgartner@st.oth-regensburg.de	July 20, 2020	Has site access	Show details ⋮

Figure 5. Access authorization of individual users to confluence spaces/pages.

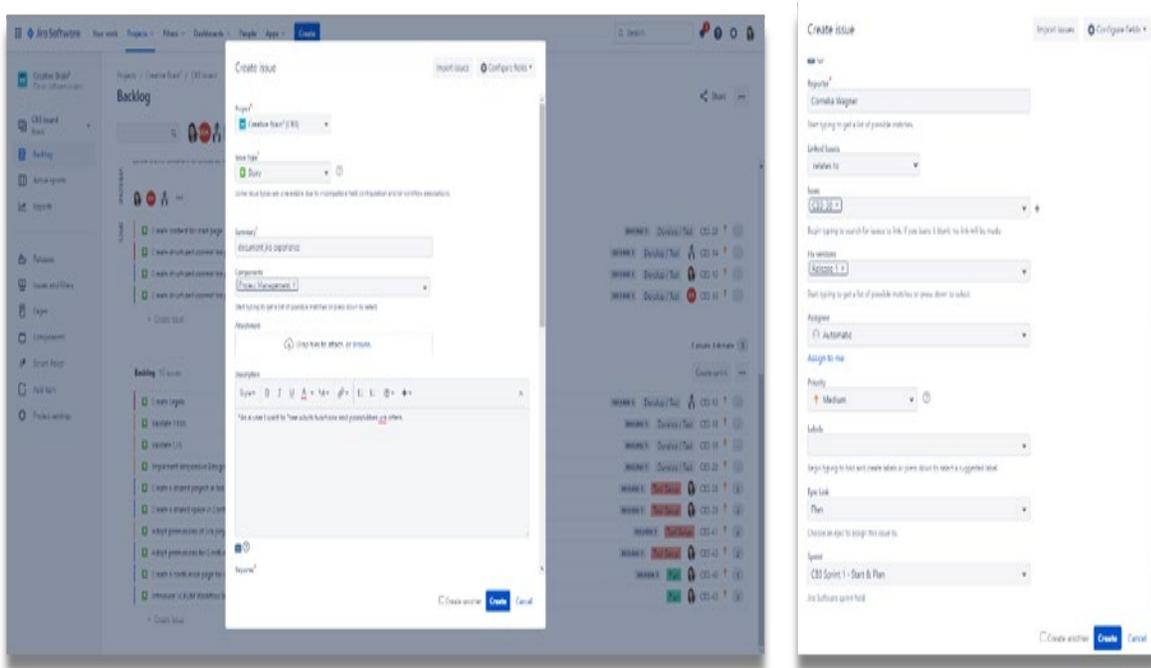


Figure 6. Creation of a new issue

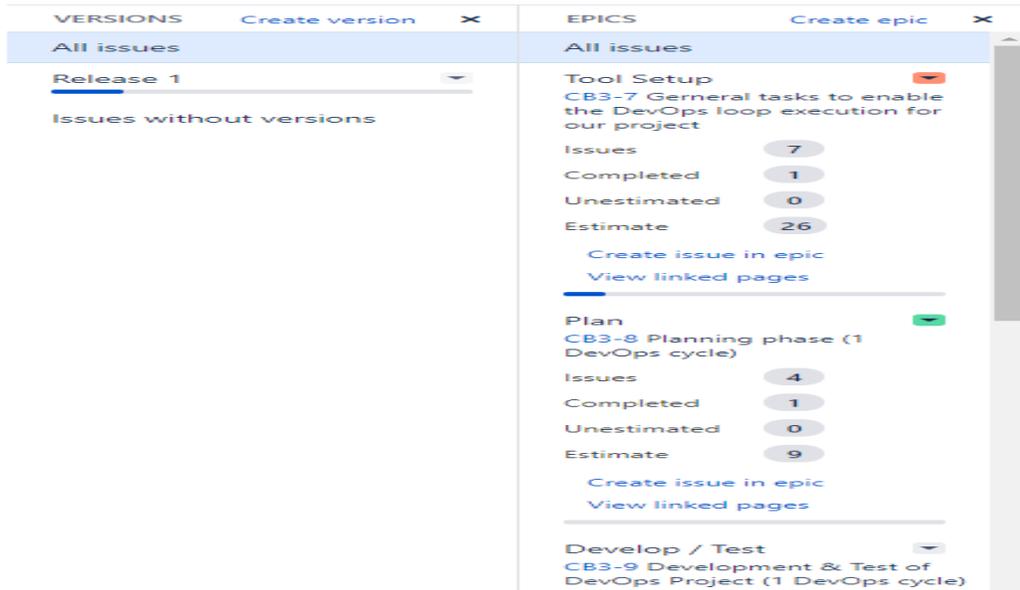


Figure 7. Structure the issues with versions

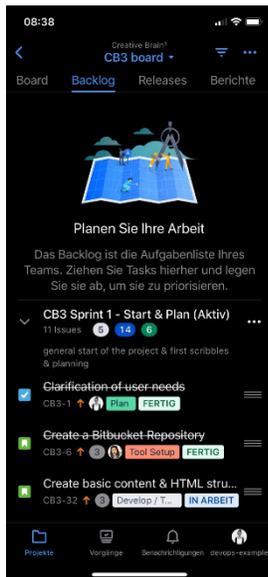


Figure 8. Backlog view.

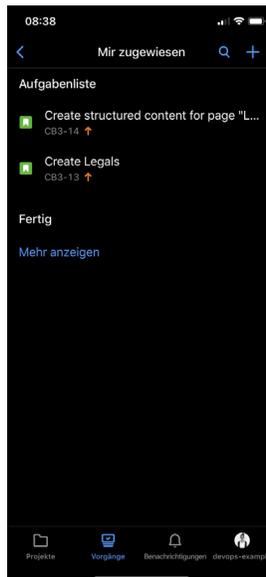


Figure 9. One member's assigned tasks.



Figure 10. Notification of changes/progress



Figure 11. Release of features in Jira.

Issues in Information Systems

Volume 22, Issue 4, pp.117-133, 2021

The screenshot shows a Jira release page for 'Creative Brain' with the version 'Release 1' marked as 'UNRELEASED'. A 'Release' button is visible in the top right. Below the header, there are statistics: 0 Warnings, 20 Issues in version, 4 Issues done, 2 Issues in progress, and 14 Issues to do. A table lists issues with columns for priority, type, key, and summary. On the right, a table shows assignees and their status: Laura Baumgartner (DONE), Cornelia Wagner (DONE), and Laura Baumgartner (DONE, MERGED).

P	T	Key	Summary	Assignee	Status	Development
↑	🔴	CB3-1	Clarification of user needs	Laura Baumgartner	DONE	
↑	🟢	CB3-6	Create a Bitbucket Repository	Cornelia Wagner	DONE	
↑	🟢	CB3-12	Create a Background Image	Laura Baumgartner	DONE	MERGED

Figure 12. Release of features visible in Jira.

The screenshot shows a Git repository interface on the left and a terminal window on the right. The repository view shows a file tree with folders 'assets' and 'images', and files '.gitignore', 'LICENSE.txt', 'README.txt', and 'index.html'. The terminal window shows the following commands and output:

```
MINGW64/c/Users/laura/team-website
laura@DESKTOP-54DI8I5 MINGW64 ~/team-website (master)
$ git push
Everything up-to-date
laura@DESKTOP-54DI8I5 MINGW64 ~/team-website (master)
$ git commit -m "Add Template Mask"
*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

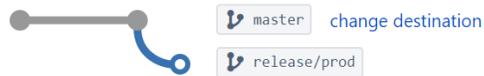
to set your account's default identity.
Omit --global to set the identity only in this repository.
fatal: unable to auto-detect email address (got 'laura@DESKTOP-54DI8I5')
laura@DESKTOP-54DI8I5 MINGW64 ~/team-website (master)
$ git config --global user.email "laura.baumgartner@st.oth-regensburg.de"
laura@DESKTOP-54DI8I5 MINGW64 ~/team-website (master)
$ git config --global user.name "Laura Baumgartner"
laura@DESKTOP-54DI8I5 MINGW64 ~/team-website (master)
$ git commit -m "Add Template Mask"
[master 2edd732] Add Template Mask
 55 files changed, 14518 insertions(+)
```

Figure 13. Adding the HTML template to the repository.

CB3 Website / Team Website / Team Website / Branches

release/prod

Check out ▾ View source Merge ⋮



No restrictions edit

Check out your branch

This branch does not contain any changes — check it out on your local machine to do some work.

Check out in Sourcetree

You can also use this command to check out your branch:

```
git fetch && git checkout release/prod
```

Figure 14. Creating a new branch to work on. The master files will not be modified.

Pull requests

Your pull requests

Activity

Reviewers

Builds

index.html edited online with Bitbucket - Google Analytics [CB3-17] → master

Laura Baumgartner - #13, 4 days ago in Team Website



Jira Software issues

devops-example ▾

Issue	Status	Assignee	Actions
CB3-37 Create legals	FERTIG		⋮
CB3-12 Create a Background Image	FERTIG		⋮
CB3-14 Create structured content for page "Laura"	ZU ERLEDIGEN		⋮
CB3-13 Create Legals	ZU ERLEDIGEN		⋮

Figure 15. The Dashboard of Bitbucket while already working on a project.

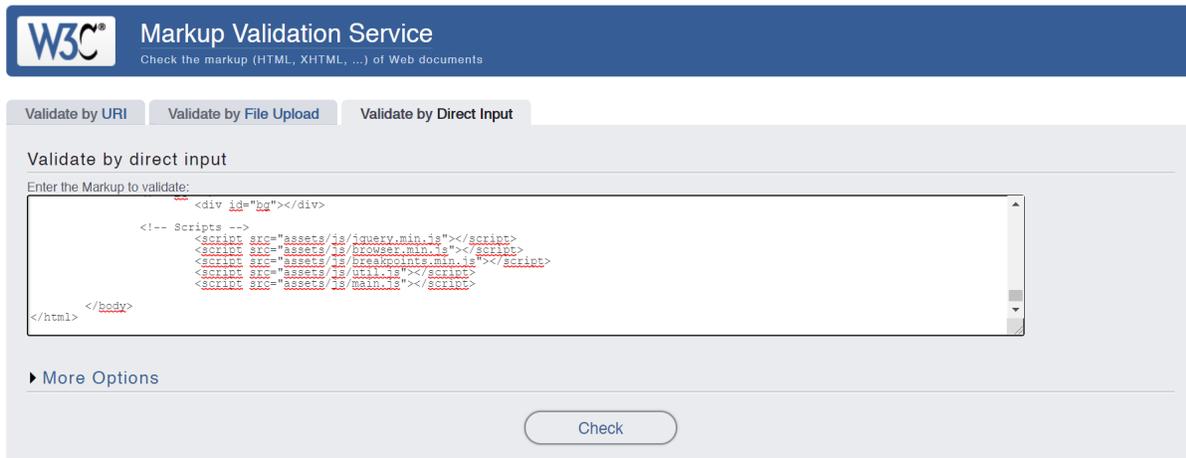


Figure 16. The W3C Validation Service - Direct Input (1/3).

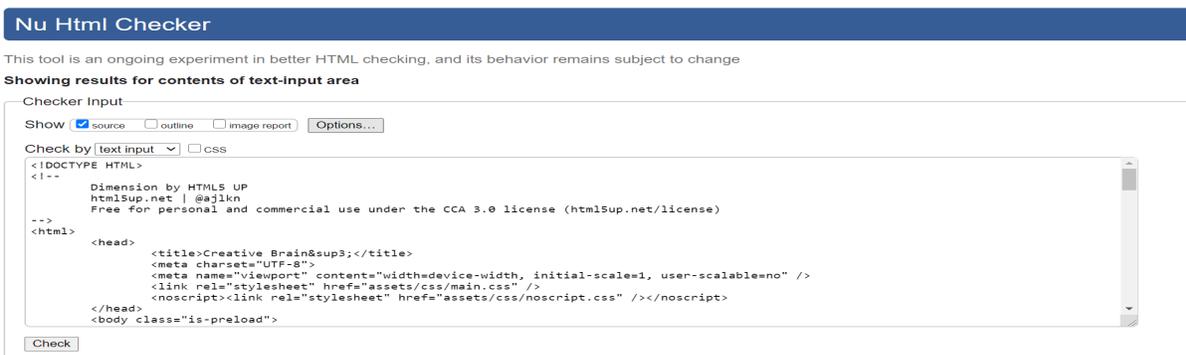


Figure 17. The W3C Validation Service - Direct Input (2/3).

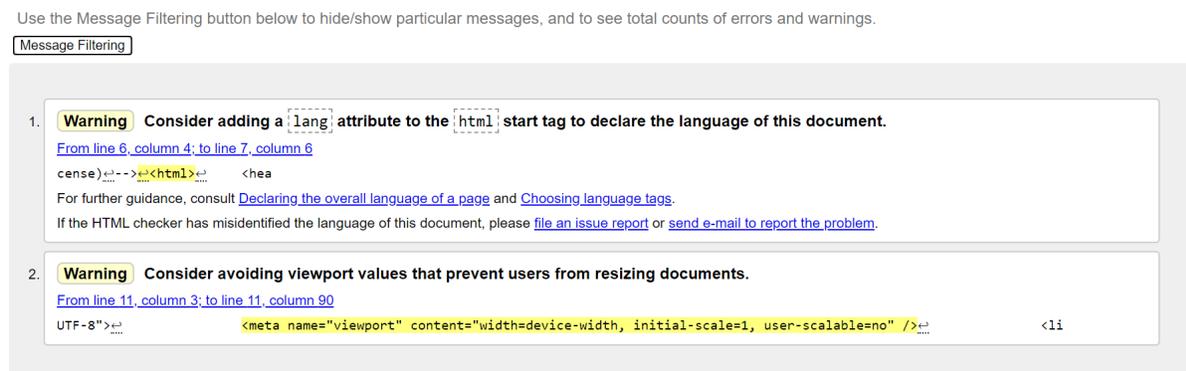


Figure 18. The W3C Validation Service - Direct Input (3/3).

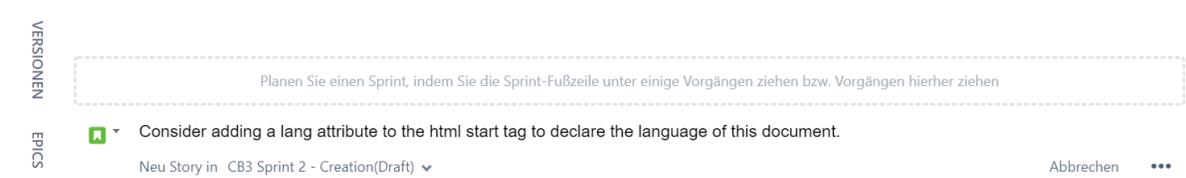


Figure 19. Writing a new Backlog story in Jira with the W3C recommendation.

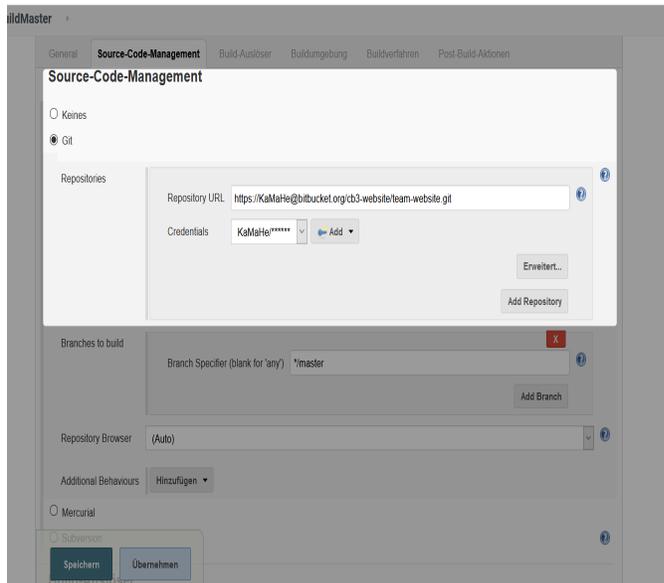


Figure 20. Creating a Jenkins Job (1/2).

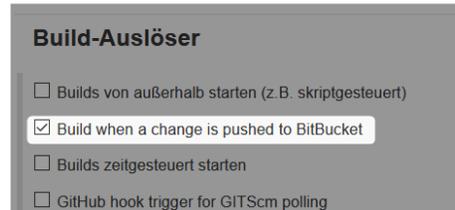


Figure 21. Creating a Jenkins Job (2/2).

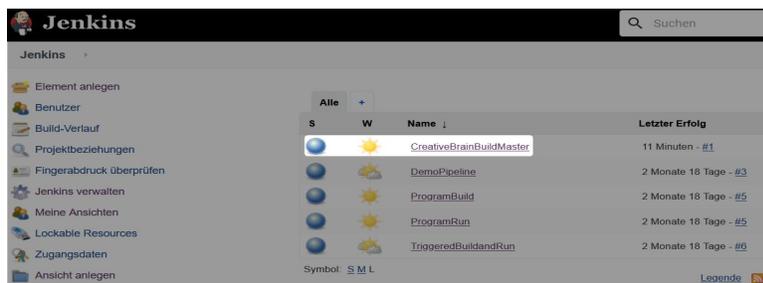


Figure 22. Successful run after push.

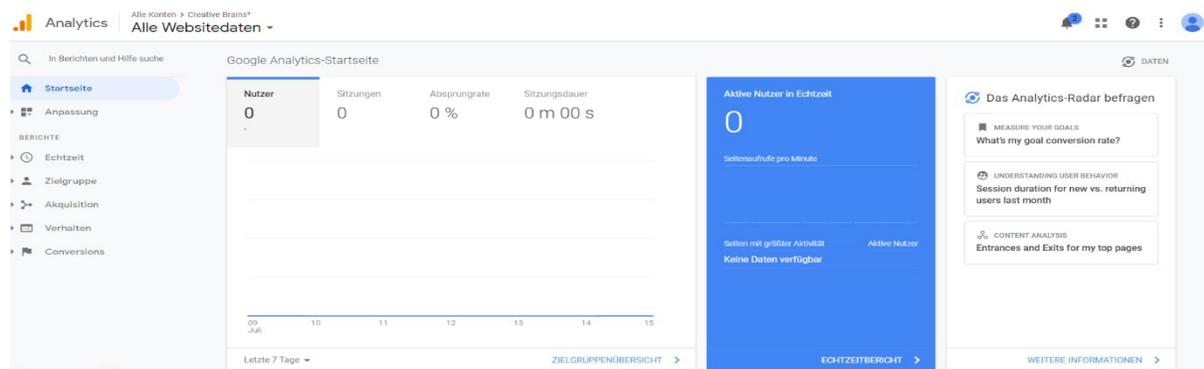


Figure 23. Google Analytics start page: Overview about important KPIs.

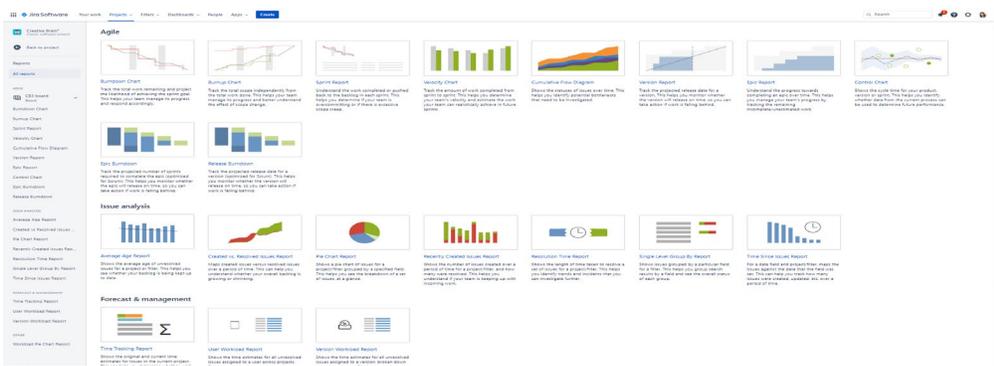


Figure 24. Jira agile reports - monitoring of development itself.

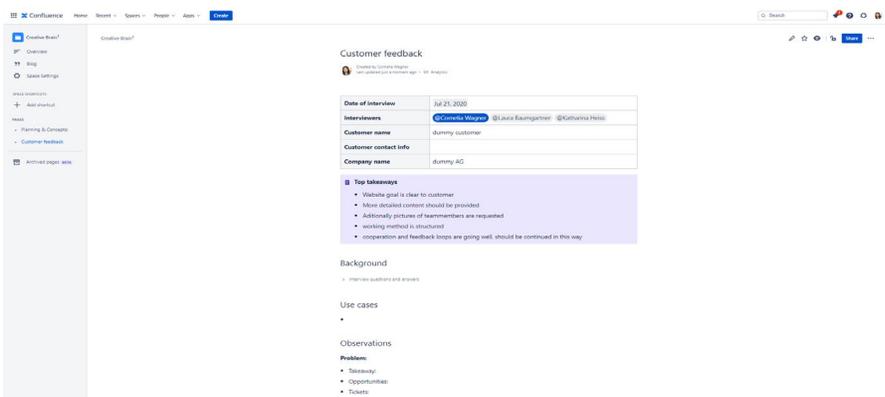


Figure 25. documentation of customer feedback (interview) using a Confluence template.