Seminar Report

An Evaluation of Ticketing Systems

Jakob Hampel

MatrNr: 13326243

Supervisors: Stefanie Mühlhausen, Jonathan Decker, Julian Kunkel

Georg-August-Universität Göttingen
Institute of Computer Science

March 30, 2024

Abstract

Allowing for efficient communication to take place, IT service providers commonly make use of Service Desk Systems (SDS), mostly in the form of so-called *ticketing systems*. Over time, these systems have become an essential tool for software development, issue tracking and customer support. While the handling of incoming requests is the common ground of all ticketing systems, there are substantial differences with respect to the use case and the range of possibilities is rather large. This term paper aims at giving an introduction to the fundamentals of ticketing systems and reports the hands-on experience of using five different systems. Furthermore, the five systems are evaluated based on various criteria and the eligibility for different use cases is determined for each of them. As Machine Learning (ML) integration is continuously evolving, the automation of ticketing systems is a topic of active research and discussion in the scientific literature and its recent advances are presented in this term paper as well.

Statement on the usage of ChatGPT and similar tools in the context of examinations

In this work I have used ChatGPT or a similar AI-system as follows:

□ Not at all

☑ In brainstorming

□ In the creation of the outline

□ To create individual passages, altogether to the extent of 0% of the whole text

□ For proofreading

□ Other, namely: -

I assure that I have stated all uses in full.
Missing or incorrect information will be considered as an attempt to cheat.

Contents

List of Figures

List of Abbreviations

AI	Artificial Intelligence
FAQ	Frequenty Asked Questions
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
ML	Machine Learning
OTRS	Open Ticket Request System
SaaS	Software-as-a-Service
SDS	Service Desk Systems
SVM	Support Vector Machine

1 Introduction

In today's rapidly changing market environment, IT service providers cannot underestimate the importance of quality assurance, customer satisfaction and constant monitoring of their products. Over the years, the Information Technology Infrastructure Library (ITIL) has become the de-facto standard for Information Technology Service Management (ITSM), providing a collection of best practices and defining important management processes that are globally recognised [ZGG09; IAG16]. A crucial part of successful ITSM is the deployment of Service Desk Systems (SDS), as they serve as a single point of contact between users and providers. Although the concept of service desks is certainly not a novel one, there are numerous advantages of modern service solutions over traditional communication strategies. An early example can be found in the case study of Foo et al. [Foo+00], which describes the implementation of a web-based SDS, thereby replacing the previous system, which was mainly characterized by long-distance telephone calls. The authors identified the old system as ineffective due to various factors, namely: slow communication, hence time-consuming and costly, inconvenience for customers, few options to prioritize incoming requests, repetitive inquiries, no (system-integrated) possibility to share knowledge from previous customer interactions and no possibility to automate the system. Given the increasing popularity of the world wide web, the migration to web-based SDS became more and more important. Among them, bulletin board systems are the earliest form, providing a collection of Frequenty Asked Questions (FAQ) [Foo+00]. Although FAQs are still a relevant tool in customer support, more sophisticated methods have emerged to provide reliable communication ways, resulting in a large variety of modern SDS that are based on so-called *ticketing systems* with capabilities beyond the scope of simple customer support.

In contrast to traditional SDS approaches, a ticketing system enables all its users to create requests at any time, thereby resolving the dependency on predefined service hours. As a single point of contact, the ticketing system can then bring together the numerous communication channels that are nowadays employed, such as e-mails, websites, contact forms, social media or the SDS itself. Any request sent is stored as a *ticket* that holds the necessary information associated with it. Although the exact information to be stored depends on the SDS implementation, a ticket commonly is characterized by a title, a description, a status, a priority, a category and a unique identifier [FDW22; MLM22]. Tickets can then be used to store various types of support requests, such as issues, incidents, questions or tasks to be solved.

Just like the specific "inventory" of a ticket, its life cycle depends on the SDS implementation but is commonly structured as follows: A user makes a request using one of the provided communication channels, entering at least a ticket title and a description, thereby effectively *creating* a ticket, which is then displayed in the SDS for the ITSM team, the so-called *agents*, to recognize it. The ticket can then be *assigned* to a member of the IT staff, who may e.g. answer the ticket creator or fix an associated issue. In case this interaction yields some kind of successful outcome, the assigned worker may initiate a ticket *resolution*, thereby marking the ticket as "solved". Alternatively, the agent may identify the ticket as being already solved, or irrelevant. In any case, the ticket is then *closed*. In case the associated issue, question or task cannot be solved by the worker, he or she may *escalate* the ticket, which then brings it up to a higher level in the IT staff hierarchy (which again depends on the SDS implementation and the organization within the system's users). Ticket escalation hence serves as an important tool to hand over tasks that may exceed a worker's capabilities. While ticketing systems may vary with respect to which user can access and edit the different parts of each ticket, the ultimate goal in any case is fast ticket closure and the efficient processing of incoming requests [GK19; RKK18].

Having outlined the rough structure of ticketing systems, its advantages over traditional SDS become visible. As the communication between inquirer and respondent is done in an asynchronous manner, requests can be managed much more efficiently. Tickets can be easily overviewed, sorted, ordered, grouped and filtered. On the other hand, the risk of overlooking a request is minimized, since each ticket remains present as long as it is not explicitly closed. Serving as a single point of contact, all communication channels are equally treated and responders can work on each request more efficiently without worrying about missing out on a specific channel. The desired result is increased customer support or in general, better management of incoming requests, while reducing the need for time-consuming human interventions [TM16]. Moreover, previously resolved tickets can be used to reduce the time needed when similar requests are submitted, thereby allowing knowledge sharing. Montgomery et al. [MLM22] explicitly point out that, by sharing the knowledge from previous customer interactions, the "evolutionary refinement of the issues" is a significant advantage of using ticketing systems. Lastly, the continuous improvement of ITSM is not to be neglected [IAG16; Jän12] and ticketing systems provide many possibilities for monitoring their performance. One of the most important ones is the time needed for request resolution [Eck10] and most state-of-the-art ticketing systems provide even more performance measures, such as the ticket closure rate in a given time window or the number of interactions needed until ticket resolution. Concluded, ticketing systems sought to enhance the workflow in day-to-day interactions and have become an indispensable tool in the ITSM environment.

Depending on the specific use case, ticketing systems may be used in various ways, ranging from customer support over issue tracking for a software in development to comprehensive project management. Although they all serve as a communication tool, the workflow, the ticket creators and the information that is shared through tickets differ for each use case. In the following, I will introduce three of them.

Regarding *customer support*, a ticketing system may be used for the communication between a company offering a product and their customers. In this scenario, the integration of various communication channels plays a key role, as the ticketing system is intended to serve as a single point of contact, while offering the customer many possibilities to get in contact with the company, e.g. email, or social media. In case of arising questions, problems with the product or the product distribution, or other types of customer requests, the support team can then efficiently respond to all tickets in an organized way and prioritize more serious incidents. Importantly, the agents in the ticketing system may not only be IT experts, but instead stem from the customer support team. Using ticket escalation, problems can then be handed over to the experts. Here, the time frame between ticket creation and delivery of service is of crucial value as it mainly determines customer satisfaction [IAG16].

Regarding *issue tracking*, a ticketing system may be used for software products. Again, the ticket creators in this case are the users/customers of the software while the company that offers the software manages the incoming tickets. A ticket may hold much more information than a simple customer question and could refer to specific categories or details of the software, such as the fix version [MLM22]. Typically, the agents in the system then are IT experts, as they have to deal with more sophisticated requests. Therefore, issue tracking is an important part especially for open-source software, as users actively help in improving a software product, hence making the ticketing system a tool for continuous development.

Regarding *project management*, there are numerous tools for efficient task distribution and monitoring. Among them, ticketing systems are widely spread, as they provide an efficient way to keep track of ongoing tasks, issues to resolve and important things to do. A reliable way to communicate within a team is one of the most important desired outcomes, since it is widely seen as a main factor that determines project success or failure [PM90]. In contrast to the other use cases, here both the ticket creators and the agents typically stem from the same group, as the ticketing system is rather used as a collaborative working tool. Here, the efficient allocation of tickets to the different users in the system is a key factor, as well as the possibility to create reports and measurements of the user's performance. Moreover, a hierarchical structure among the users is commonly deployed.

The goal of this term paper is to explore different ticketing systems and provide an overview about the aspects that can determine whether or not a specific system can offer advantages. Apart from the use cases that were just described, the decision for or against a ticketing system depends on numerous factors. In general, deploying a ticketing system to substitute a traditional SDS typically should be cost-effective, by saving time. This has been demonstrated in various case studies. Foo et al. [Foo+00] report a substantial cost reduction after implementing a web-based help desk system at a university in Singapore. Yamaoka et al. [Yam+19] also report a significant improvement in response time of the IT staff at the Kyoto Institute for Technology after using a ticketing system to receive requests from students. Moreover, the authors mention that it was possible to estimate the workload for support requests much easier than before. In the following sections, I will present a subset of the ticketing systems currently available on the market and state for which use cases each of them is suitable.

2 Methods

2.1 Procedure

Although ticketing systems are increasing in popularity, there is an imbalance in software engineering research about them [MLM22], making a comparison somewhat challenging. Instead of focusing on scientific literature and review papers only, I also explored the free trials provided by practically every ticketing system on the market. Most of the time, these free trials are limited to a fixed number of agents, or a fixed number of days, where interested persons can try out the user interface and get comfortable with the overall structure. Nevertheless, a main challenge was to find ticketing systems that provide comparable interfaces and workflows, despite them being targeted at different audiences. However, for the scope of this term paper, free trials were sufficient to get insight into the advantages and disadvantages of each system and to draw conclusions about the suitability for each use case covered in the introduction.

2.2 Selected Systems

Jira

Jira [1] is a web-based SDS by Atlassian and has established itself as one of the most popular tools on the market [MLM22]. Importantly, Jira was initially intended for software development, but has become interesting for various professional fields that may not necessarily be IT-related. While Jira is web-based, Atlassian also offers a version with its own data center or Jira as Software-as-a-Service (SaaS). Due to its popularity, but also due to its large functionalities, I decided to select Jira for this comparison. After setting up three different Atlassian accounts, I was able to create roles for a system admin, an agent, and a user without special access rights in order to test out the system.

Znuny

The Open Ticket Request System (OTRS) [2] is a widely spread ticketing system that was formerly free software and, after 2018, split up into the free OTRS Community Edition and the chargeable OTRS platform. As part of the HPCSA course at the University of Göttingen, we explored the project Znuny [3] which provides ongoing support for the free version of OTRS. Furthermore, the free demo of the OTRS software provided me with additional information, since the interface is mainly identical to Znuny and allows an easy way to try out the basic functionality.

GitLab

GitLab [4] is a globally recognized software development tool that is used to maintain and improve IT services by a large number of companies, including the GitLab company itself [Cho+20]. Amongst other things, GitLab can be used as a ticketing system to allow efficient issue tracking for ongoing software development. Due to its relevance and its high representation in software engineering research, I included GitLab in this comparison. After creating three different GitLab accounts, I used the free trial where you can invite up to five members to a project, to explore the functionalities of using GitLab as a ticketing system.

Zendesk

Zendesk [5] is a web-based ticketing system that is especially characterized by its user-friendly interface. Targeted at users with less technical knowledge, Zendesk has become a popular tool for customer service. As a system that differs in many aspects from the other SDS in this comparison, I included it to point out the differences among ticketing

[1] https://www.atlassian.com/de/software/Jira, Last Access: 30.03.2024

[2] https://otrs.com/, Last Access: 30.03.2024

[3] https://www.znuny.org/de, Last Access: 30.03.2024

[4] https://gitlab.com, Last Access: 30.03.2024

[5] https://www.zendesk.de/, Last Access: 30.03.2024

systems. Using the free trial, I created one account representing the system admin and one account representing an agent. Creating customer accounts was not necessary, as the free trial already provides a collection of exemplary tickets and interactions.

Trac
According to their website [6] , Trac is a “minimalistic approach to web-based management of software projects”. As a simplified ticketing system, Trac provides solutions for software development and is targeted at a rather technically skilled audience, which makes it an interesting SDS for this comparison. Moreover, Trac can be used as a wiki system and, due to it being open-source, can be freely customized. Importantly, using Trac locally requires the installation of the Python language, the database management language SQL and basic command line skills, as long as one follows the official documentation [7]. After setting up a project environment and a SQLite database, I was able to run a standalone server and try out the basic features of Trac.

2.3 Evaluation Criteria

In addition to the general workflow and feeling when using any of the systems, there are some features of ticketing systems that can be used to evaluate the suitability for a specific use case. In this term paper, I will use them as criteria to provide an evaluation for each ticketing system. Some of them can be retrieved beforehand, others become visible after actively trying out the basic functionality. Note that, although I was able to get hands-on experience for each system, I had to exclude some common evaluation criteria, such as performance indicators or the long-term technical reliability, as these extend the scope of free trials and this paper.

Usability
Usability denotes the difficulty to use the system - this encompasses the user interface, the time needed to get comfortable with the system’s environment, but also the provided documentation and help for non-experienced users.

Integration Capability
Integration capability denotes the system’s possibilities to link other communication channels into the ticketing system itself. Hence, it is the driving factor to enable a single point of contact between the agents and the ticket creators.

Assessment Capability
The assessment capability of a ticketing system is determined by its possibilities to create reports, analyses, performance metrics and other types of statistics. The creation of such statistics is generally seen as very important in order to improve the workflow on an existing ticketing system [Agl+22; AB21].

Automation Capability
The automation capability of a system is characterized by the level to which various processes can be automatized. This is especially important for agents that have to deal with

[6] https://trac.edgewall.org/, Last Access: 30.03.2024
[7] https://trac.edgewall.org/demo-1.4/wiki/TracGuide, Last Access: 30.03.2024

a large number of incoming tickets.

Scalability
Scalability to the future denotes the opportunities of extending the system to handle a larger amount of e.g. agents, tickets or functions. This may take place in the form of predefined upgrades, given by the system provider, or in the form of open-source software that can be freely extended.

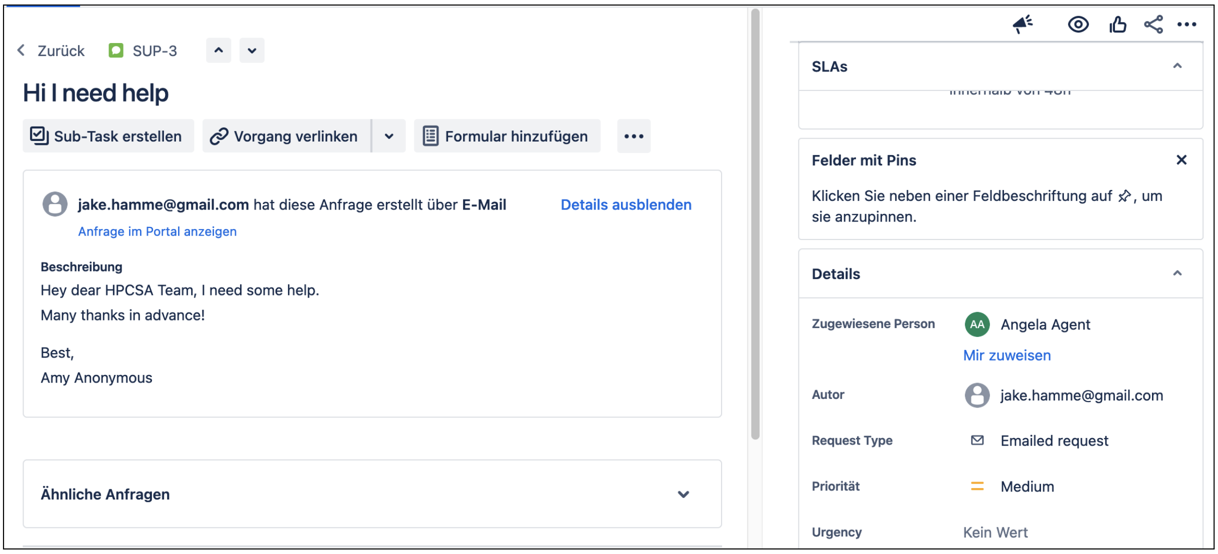
Capacity and Cost
The capacity of a ticketing system is mainly characterized by the number of agents that can work on it. Most of the time, it is intertwined with the charging fee of the system or the selected pricing plan. In contrast, some ticketing systems may not require any charging fee at all.

3 Evaluation

3.1 Separate Evaluation

Jira
When starting to work with Jira, I decided on “Service Management” as a project type, however, there are various other types and views that can be chosen from. A custom email address and an own website is immediately created when the service desk is opened for the first time. Ticket creation is then possible by invoking the website, as can be seen in figure 1 or by sending an email, where the subject of the email is used as the ticket’s title and the email content as the ticket’s description. A ticket is then visible in the agents ticket queue, as can be seen in figure 2, as well as in customizable issue boards (figure 3). I tried both ways to create a ticket and each time, the corresponding agents were notified by their respective email address. Similarly, ticket assignment was possible by using the admin user and the respective agent was again notified per email. Jira also offers numerous ways to create performance statistics, as can be seen in figure 4 and allows the customization of the already mentioned email notifications. In general, I experienced the amount of functionalities as slightly overwhelming, however, as the user interface is carefully designed, getting to know the system’s functions is definitely possible. Nevertheless, it requires a decent training period, which I experienced as challenging and was also pointed out in a case study by Yamaoka et al. [Yam+19]. The integration of multiple channels is possible as well: In addition to the custom email address and website, Jira can link a project’s tickets to Slack and Microsoft Teams. However, integrating social media channels such as LinkedIn is currently not possible. Jira also provides various ways to automate a project’s workflow, from ticket assignment over warning messages to tools for increased customer engagement. Automation with AI is not possible in the free version, but can be activated when switching to a higher pricing plan. Lastly, due to its large range of functions, Jira offers high scalability to the future.

Znuny
The user interface of Znuny is very simplified and easy to grasp, although it does not meet up to the standards of modern user interfaces. When creating a ticket, users can do so on



< Zurück SUP-3

Hi I need help

jake.hamme@gmail.com hat diese Anfrage erstellt über **E-Mail** Details ausblenden

Anfrage im Portal anzeigen

Beschreibung

Hey dear HPCSA Team, I need some help.

Many thanks in advance!

Best,

Amy Anonymous

Ähnliche Anfragen

SLAs

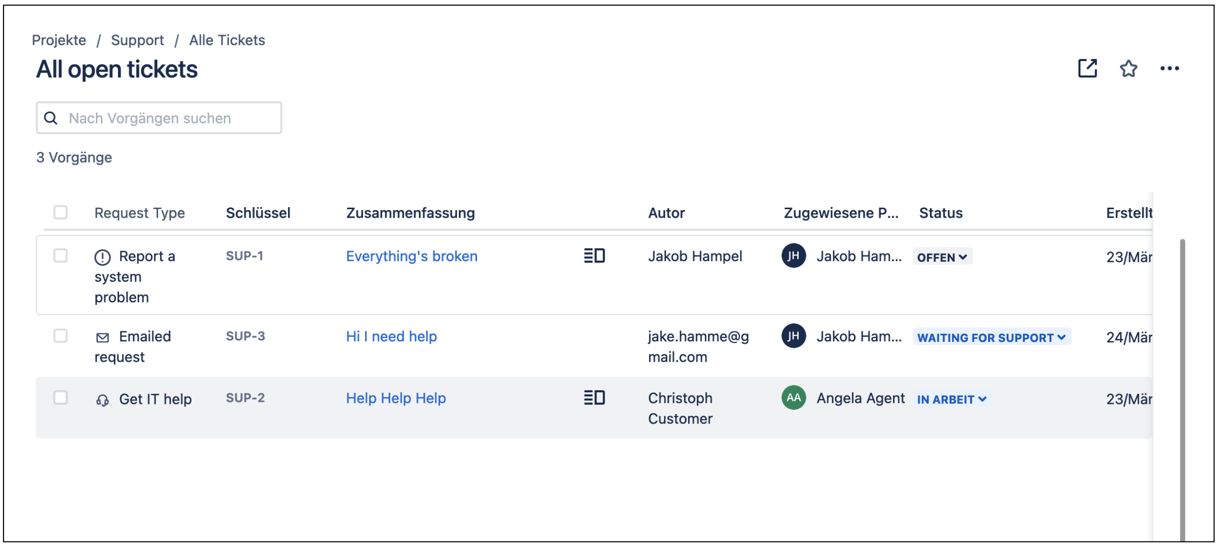
Felder mit Pins ✕

Klicken Sie neben einer Feldbeschriftung auf 📌, um sie anzupinnen.

Details

Zugewiesene Person	AA Angela Agent Mir zuweisen
Autor	jake.hamme@gmail.com
Request Type	Emailed request
Priorität	= Medium
Urgency	Kein Wert

Figure 1: The interface of Jira when creating a ticket.



Projekte / Support / Alle Tickets

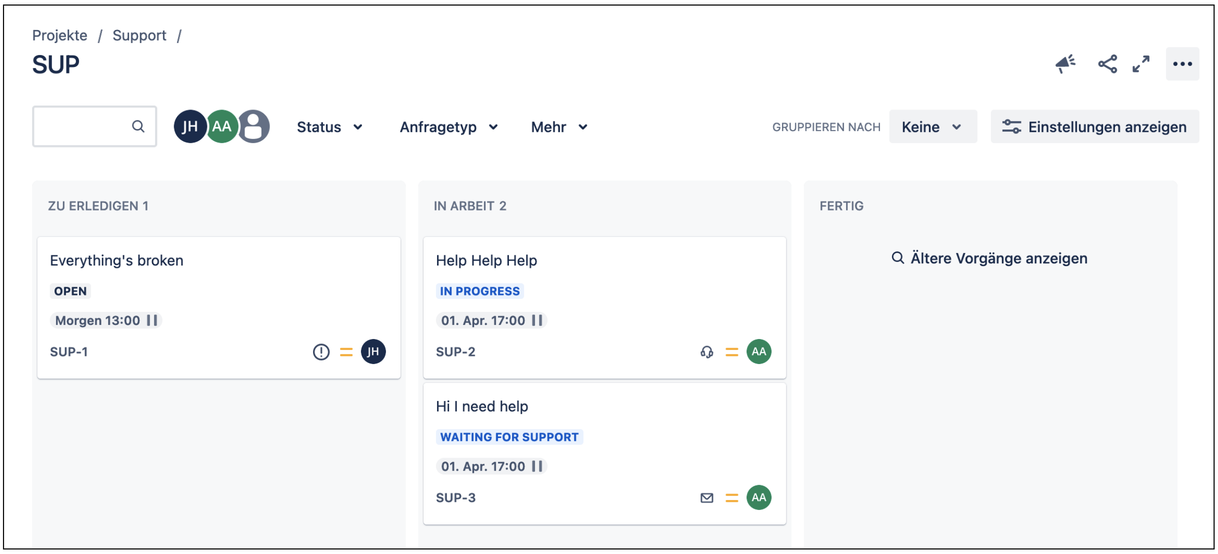
All open tickets

3 Vorgänge

□	Request Type	Schlüssel	Zusammenfassung		Autor	Zugewiesene P...	Status	Erstellt
□	Report a system problem	SUP-1	Everything's broken	≡◫	Jakob Hampel	JH Jakob Ham...	OFFEN	23/Mär
□	Emailed request	SUP-3	Hi I need help		jake.hamme@gmail.com	JH Jakob Ham...	WAITING FOR SUPPORT	24/Mär
□	Get IT help	SUP-2	Help Help Help	≡◫	Christoph Customer	AA Angela Agent	IN ARBEIT	23/Mär

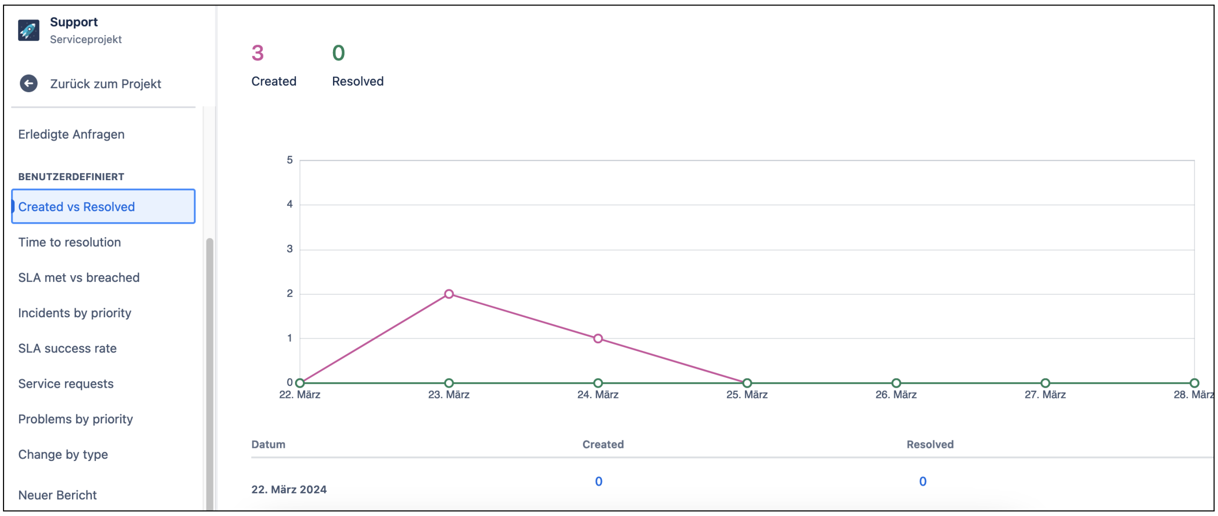
Figure 2: The ticket queue in Jira.

the website itself and allocate it to a postbox, thereby categorizing the ticket, as can be seen in figure 5. Agents can then see that ticket and assign it to another agent or modify the ticket itself, as can be seen in figure 6. Each answer to and modification of a ticket is visible in an overview page (compare to figure 7), moreover, the ticket requesters, or the customers, are listed in an extra page (compare to figure 8). In general, the rudimentary design makes Znuny a very easy-to-use tool but does not provide a wide range of functionalities. However, it is possible to generate reports and simple statistics over opened and closed tickets. Znuny also provides extensions for including more communication channels, although social media integration is not possible and the number of channels is fairly limited. Similarly, Znuny allows the linking of predefined processes and tickets to move towards automated workflows.



ZU ERLEDIGEN 1	IN ARBEIT 2	FERTIG
Everything's broken OPEN Morgen 13:00 SUP-1 JH	Help Help Help IN PROGRESS 01. Apr. 17:00 SUP-2 AA	
	Hi I need help WAITING FOR SUPPORT 01. Apr. 17:00 SUP-3 AA	

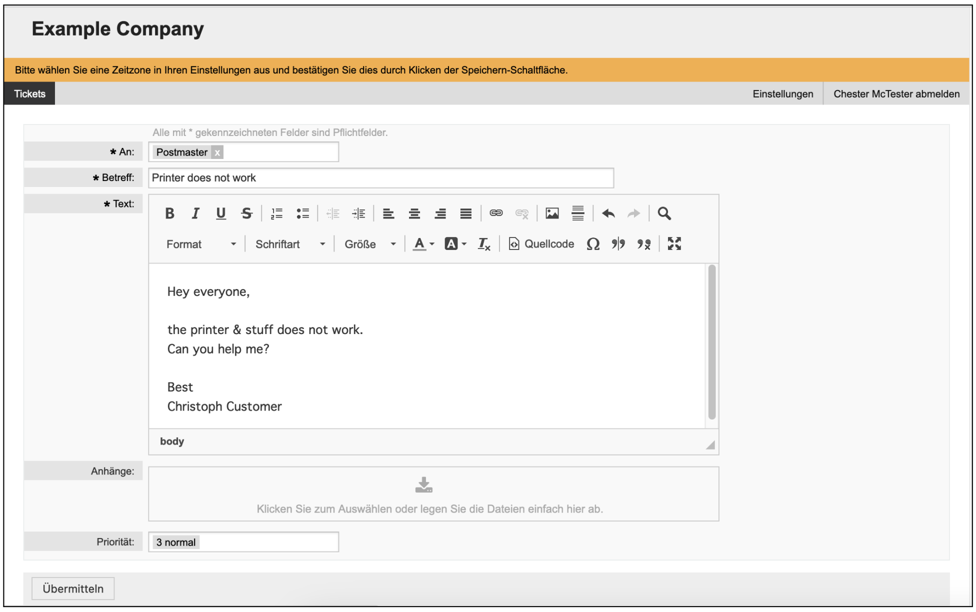
Figure 3: The issue boards in Jira.



Datum	Created	Resolved
22. März 2024	0	0

Figure 4: One of the statistical reports in Jira, here ticket performance over days.

GitLab
GitLab's ticketing system usually is embedded into a repository management and is hence targeted at users that want to maintain or develop software, as code integration is easily possible. Users can create both issue tickets and incident tickets, using the repository's website, as can be seen in figure 9, or by writing to the repository's custom email address (although this way is rather unintuitive). Apart from that, the integration capability of GitLab's ticketing system is rather limited. Each time a ticket was created or altered, the corresponding GitLab accounts were notified by email and the tickets could be clearly overviewed in GitLab's ticket boards (compare to figure 10). Different user roles define the access rights of each user and can be useful for project management. Moreover, GitLab provides a wide range of performance measures and analyses for tickets, however, automation capabilities are fairly limited. In general, the user interface of GitLab is designed in a clear manner and has the advantage of being well-known among software developers.



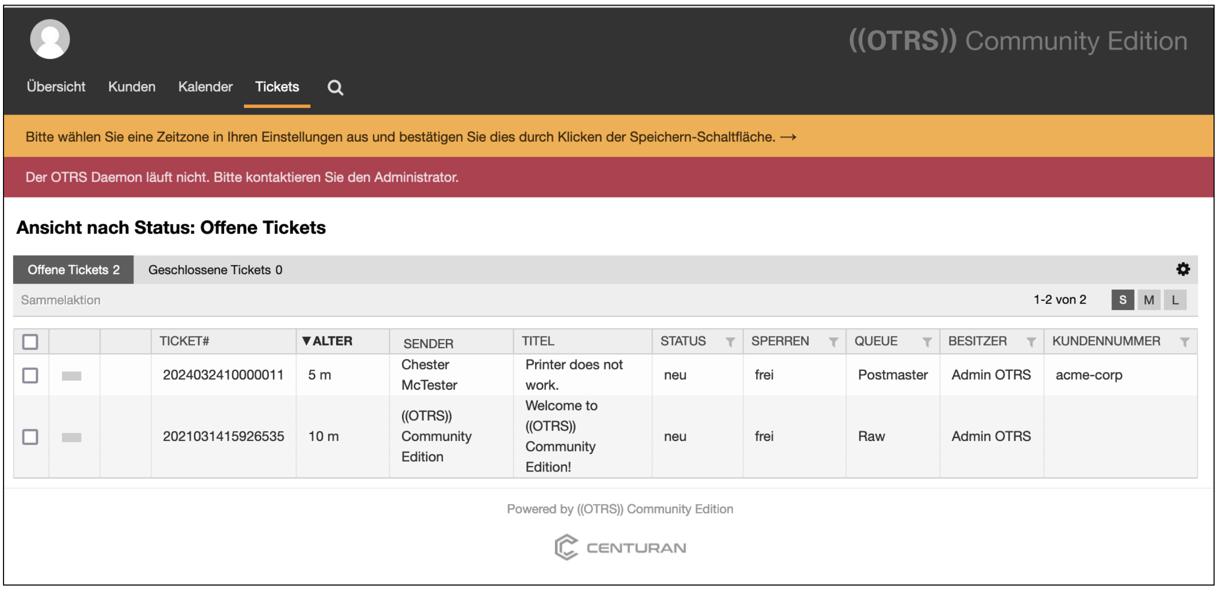
Example Company

Bitte wählen Sie eine Zeitzone in Ihren Einstellungen aus und bestätigen Sie dies durch Klicken der Speichern-Schaltfläche.

Alle mit * gekennzeichneten Felder sind Pflichtfelder.

* An:	Postmaster
* Betreff:	Printer does not work
* Text:	Hey everyone, the printer & stuff does not work. Can you help me? Best Christoph Customer
Anhänge:	Klicken Sie zum Auswählen oder legen Sie die Dateien einfach hier ab.
Priorität:	3 normal

Figure 5: The ticket creation in Znuny.



Bitte wählen Sie eine Zeitzone in Ihren Einstellungen aus und bestätigen Sie dies durch Klicken der Speichern-Schaltfläche. →

Der OTRS Daemon läuft nicht. Bitte kontaktieren Sie den Administrator.

Ansicht nach Status: Offene Tickets

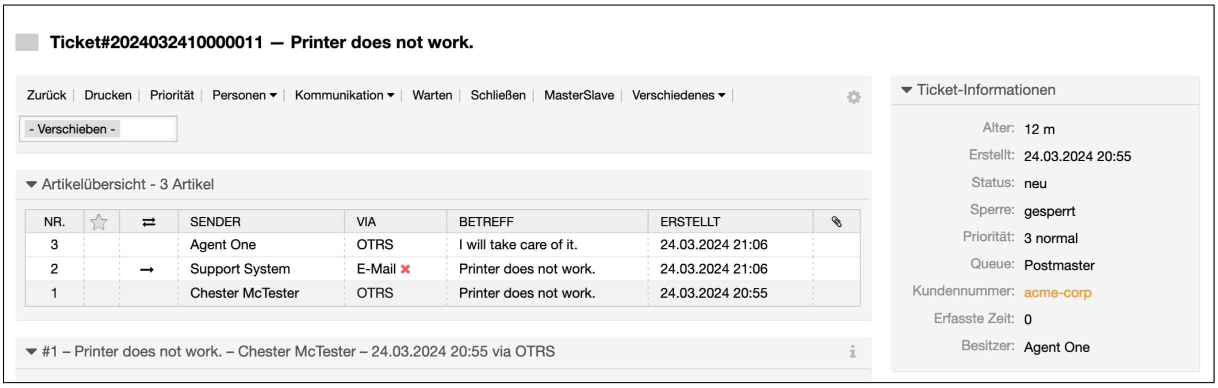
□			TICKET#	▼ALTER	SENDER	TITEL	STATUS	SPERREN	QUEUE	BESITZER	KUNDENNUMMER
□			2024032410000011	5 m	Chester McTester	Printer does not work.	neu	frei	Postmaster	Admin OTRS	acme-corp
□			2021031415926535	10 m	((OTRS)) Community Edition	Welcome to ((OTRS)) Community Edition!	neu	frei	Raw	Admin OTRS	

Powered by ((OTRS)) Community Edition

Figure 6: The ticket overview in the agent's account of Znuny.

Zendesk

Zendesk immediately presents itself as tailored for customer support when creating a project and seems to be targeted at users that bring little to no IT expertise. Setting up the project environment is as easy as can be, while providing tutorial videos and helping information at every step. Furthermore, the integration capability of Zendesk is strongly advertised in the beginning, with possibilities to select various communication channels that ought to be included in the project, as can be seen in figure 11. When creating and altering tickets, the corresponding email notifications worked as expected and the incoming tickets can be clearly overviewed in ticket queues or boards, as can be seen in figure 12. Furthermore, Zendesk provides its own platform "Zendesk Explore" to generate



Ticket#2024032410000011 — Printer does not work.

Zurück | Drucken | Priorität | Personen ▾ | Kommunikation ▾ | Warten | Schließen | MasterSlave | Verschiedenes ▾
- Verschieben -
Artikelübersicht - 3 Artikel

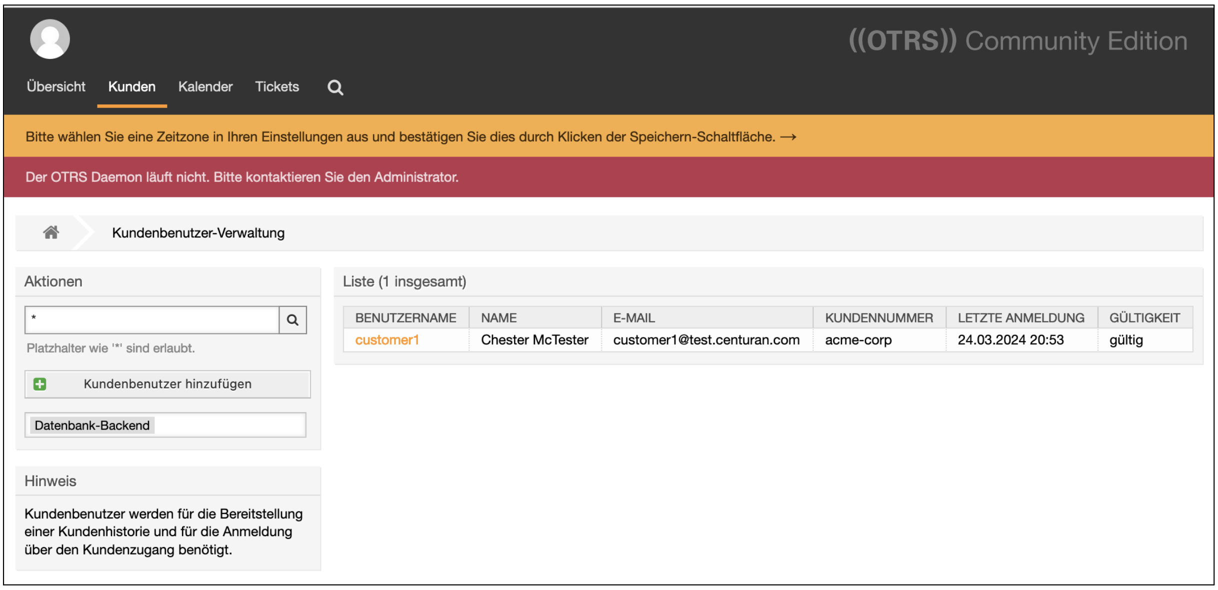
NR.	☆	⇄	SENDER	VIA	BETREFF	ERSTELLT
3			Agent One	OTRS	I will take care of it.	24.03.2024 21:06
2		→	Support System	E-Mail ✖	Printer does not work.	24.03.2024 21:06
1			Chester McTester	OTRS	Printer does not work.	24.03.2024 20:55

#1 – Printer does not work. – Chester McTester – 24.03.2024 20:55 via OTRS

Ticket-Informationen

Alter:	12 m
Erstellt:	24.03.2024 20:55
Status:	neu
Sperre:	gesperrt
Priorität:	3 normal
Queue:	Postmaster
Kundennummer:	acme-corp
Erfasste Zeit:	0
Besitzer:	Agent One

Figure 7: The history of altered tickets is visible in an overview page in Znuny.



Bitte wählen Sie eine Zeitzone in Ihren Einstellungen aus und bestätigen Sie dies durch Klicken der Speichern-Schaltfläche. →

Der OTRS Daemon läuft nicht. Bitte kontaktieren Sie den Administrator.

Kundenbenutzer-Verwaltung

Aktionen

*

Platzhalter wie '*' sind erlaubt.

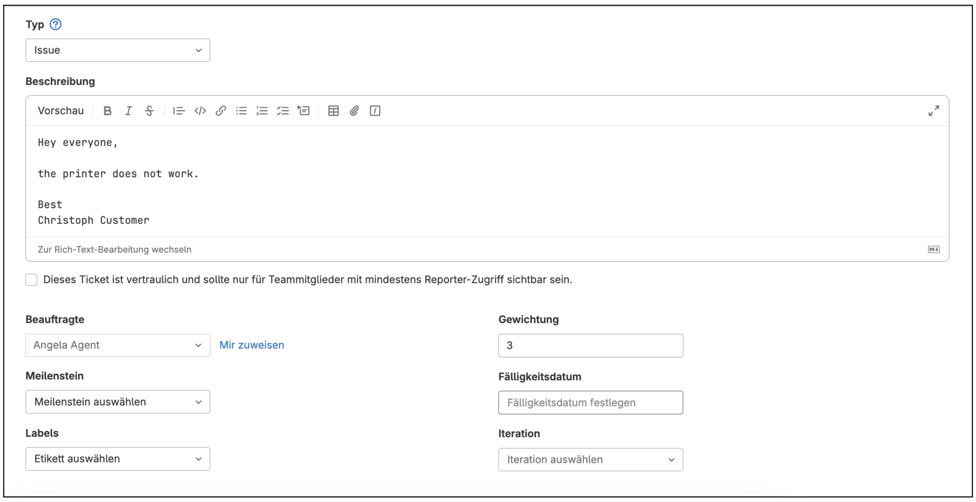
Hinweis

Kundenbenutzer werden für die Bereitstellung einer Kundenhistorie und für die Anmeldung über den Kundenzugang benötigt.

Liste (1 insgesamt)

BENUTZERNAME	NAME	E-MAIL	KUNDENNUMMER	LETZTE ANMELDUNG	GÜLTIGKEIT
customer1	Chester McTester	customer1@test.centuran.com	acme-corp	24.03.2024 20:53	gültig

Figure 8: The customer overview in Znuny.



Typ: Issue

Beschreibung

```
Hey everyone,

the printer does not work.

Best
Christoph Customer
```

□ Dieses Ticket ist vertraulich und sollte nur für Teammitglieder mit mindestens Reporter-Zugriff sichtbar sein.

Beauftragte: Angela Agent (Mir zuweisen)	Gewichtung: 3
Meilenstein: Meilenstein auswählen	Fälligkeitsdatum: Fälligkeitsdatum festlegen
Labels: Etikett auswählen	Iteration: Iteration auswählen

Figure 9: The creation of tickets in GitLab.

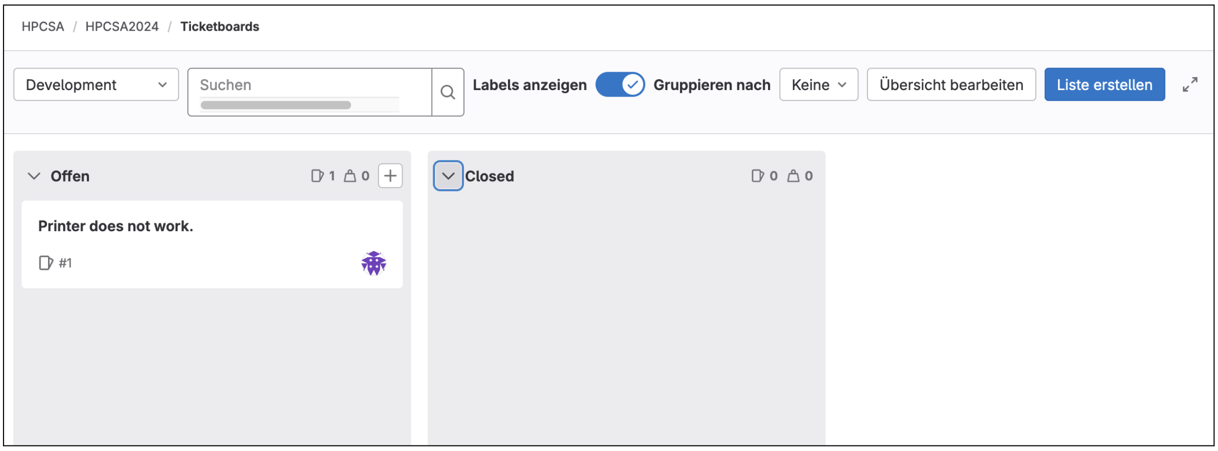
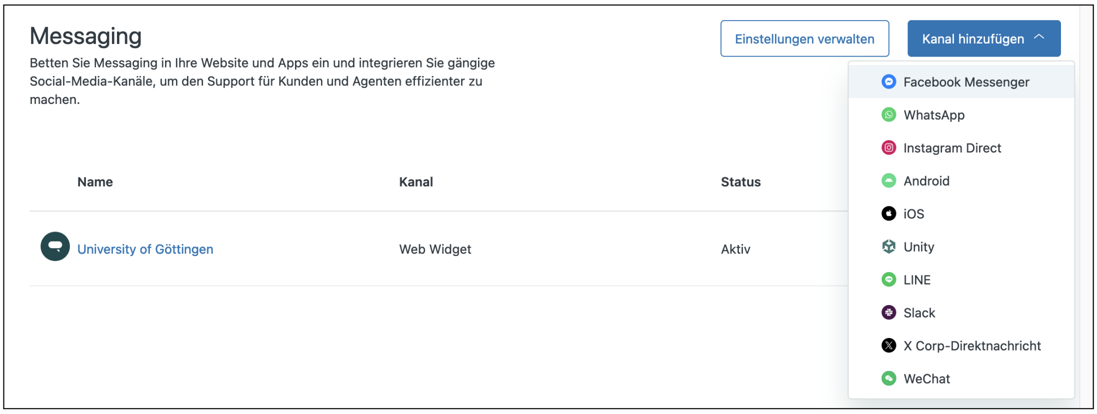


Figure 10: The ticket boards in GitLab

comprehensive reports and statistics about ticket performance and customer satisfaction. The automation capability also seems to be at the state-of-the-art, with numerous possibilities to automate processes and integrate AI. User roles can be predefined, although a sophisticated hierarchy structure does not seem to be one of Zendesk's focus aspects. In general, Zendesk offers high usability and requires very little time to get used to it but does not provide as many functionalities as Jira.


Messaging

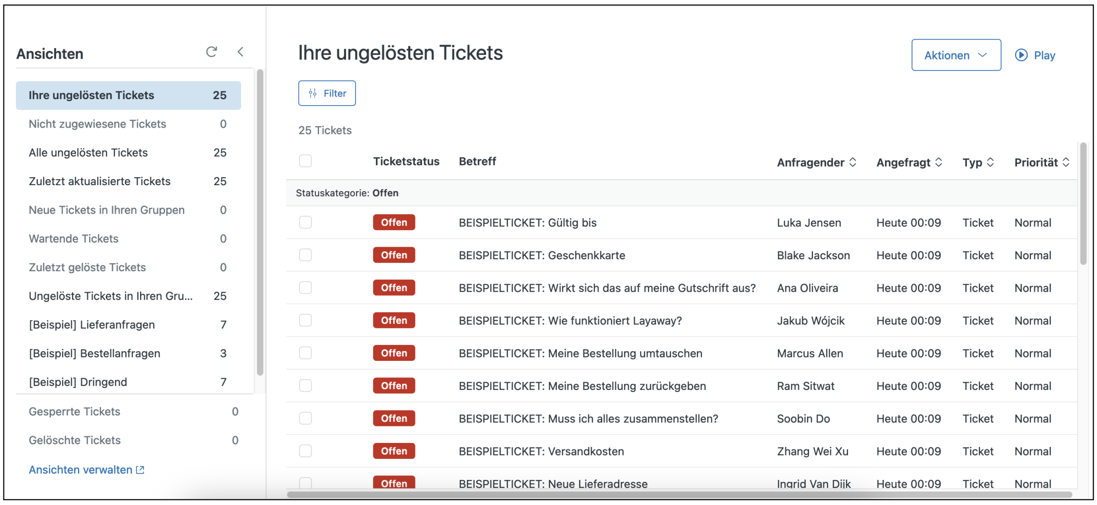
Betten Sie Messaging in Ihre Website und Apps ein und integrieren Sie gängige Social-Media-Kanäle, um den Support für Kunden und Agenten effizienter zu machen.

Name	Kanal	Status
University of Göttingen	Web Widget	Aktiv

Figure 11: The channel integration in Zendesk.

Trac

The biggest strength of Trac is its lightweight design and simplified issue tracking. On the other hand, setting up the project environment requires at least some technological expertise and took the longest, compared to the other SDS. Furthermore, the automatic spam detection seems to be a common issue in Trac, as the spam filtering is fairly strict, which made the testing phase for this system somewhat challenging. When the standalone server is set up, only authenticated users can see, create and alter tickets, whereas non-authenticated (anonymous) users can only create them, as can be seen in figure 13. The integration of other communication channels is not possible, as well as automating the ticket processes, however, as Trac is open-source, scalability to the future is definitely given and missing components can be integrated afterwards. Again, however, this is only suitable for users that bring IT expertise and a general understanding of code extensions.



Ansichten

Ihre ungelösten Tickets	25
Nicht zugewiesene Tickets	0
Alle ungelösten Tickets	25
Zuletzt aktualisierte Tickets	25
Neue Tickets in Ihren Gruppen	0
Wartende Tickets	0
Zuletzt gelöste Tickets	0
Ungelöste Tickets in Ihren Gru...	25
[Beispiel] Lieferanfragen	7
[Beispiel] Bestellanfragen	3
[Beispiel] Dringend	7
Gesperrte Tickets	0
Gelöschte Tickets	0

Ansichten verwalten

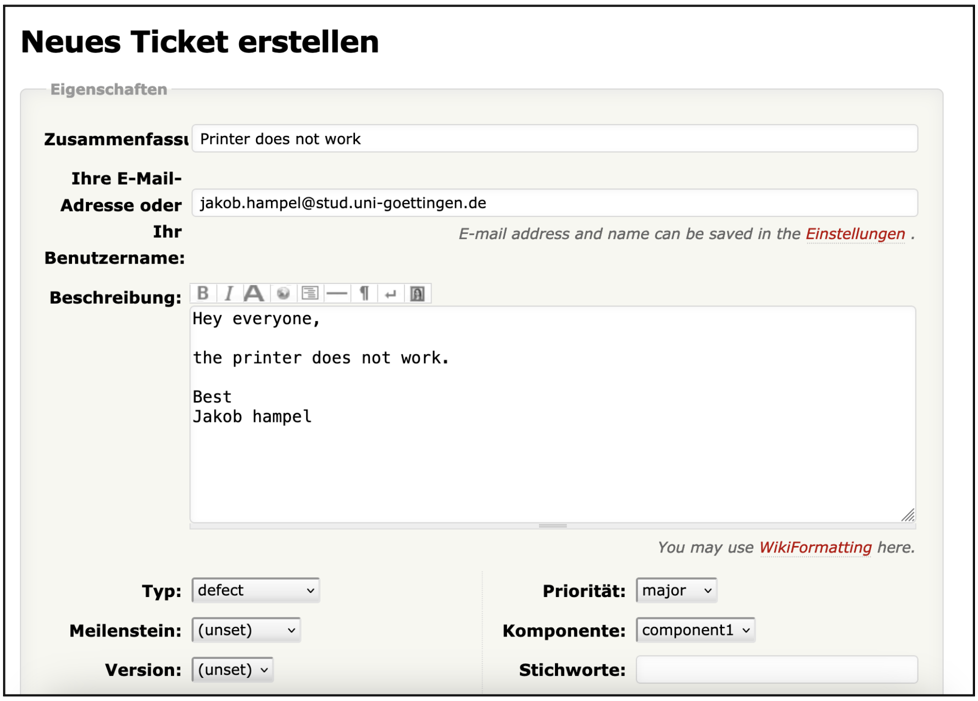
Ihre ungelösten Tickets

25 Tickets

	Ticketstatus	Betreff	Anfragender	Angefragt	Typ	Priorität
Statuskategorie: Offen						
□	Offen	BEISPIELTICKET: Gültig bis	Luka Jensen	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Geschenkkarte	Blake Jackson	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Wirkt sich das auf meine Gutschrift aus?	Ana Oliveira	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Wie funktioniert Layaway?	Jakub Wójcik	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Meine Bestellung umtauschen	Marcus Allen	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Meine Bestellung zurückgeben	Ram Sitwat	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Muss ich alles zusammenstellen?	Soobin Do	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Versandkosten	Zhang Wei Xu	Heute 00:09	Ticket	Normal
□	Offen	BEISPIELTICKET: Neue Lieferadresse	Ingrid Van Dijk	Heute 00:09	Ticket	Normal

Figure 12: The ticket overview in Zendesk.

Concluded, Trac is easy to understand and easy to use once set up, but is naturally limited in its functionalities and user-friendliness.



Neues Ticket erstellen

Eigenschaften

Zusammenfassu Printer does not work

Ihre E-Mail-Adresse oder Ihr Benutzername: jakob.hampel@stud.uni-goettingen.de

E-mail address and name can be saved in the Einstellungen .

Beschreibung:

```
Hey everyone,

the printer does not work.

Best
Jakob hampel
```

You may use WikiFormatting here.

Typ: defect

Priorität: major

Meilenstein: (unset)

Komponente: component1

Version: (unset)

Stichworte:

Figure 13: The ticket creation in Trac.

3.2 Comparison

Since each ticketing system puts its focus on different aspects, they naturally differ with respect to the evaluation criteria presented above, which allows me to point out the strengths and weaknesses in each of them. When it comes to usability, the time frame needed for getting comfortable with each system is intertwined with its functionality range. Nevertheless, the use of modern user interfaces and carefully designed documentations can substantially increase the usability, which became visible especially for Zendesk and Jira. In contrast, Trac and Znuny require very little time to get used to them, but at the same time, lack user-friendliness for technically less skilled users and provide standard documentation that may be overwhelming for starters, although Znuny's documentation offers more readability than that of Trac. GitLab poses itself in between; while requiring more time to get used to working with it, there is a wide range of support for starters, as GitLab is widely used among software developers.

Regarding integration capability, Zendesk is clearly at the top of this comparison, as the inclusion of various communication channels is highly supported and easy to use. Following up, Jira also provides decent integration capability but does not focus as much on it as Zendesk. Znuny and Trac offer far less integration capability but can be manually extended to do so. Lastly, GitLab does not set its focus on multichannel operations and has very limited possibilities to include other communication ways.

When it comes to automation capability, Jira and Zendesk provide the most possibilities to define automated processes that can enhance the user's workflow. Znuny offers rudimentary automation by predefining rules, but does not meet up to the automation level that can be achieved by Jira and Zendesk. The automation capability of GitLab and Trac is very low, but again, can be improved by manually creating extensions to the system.

The capacity and cost of each system is relatively hard to estimate, as it depends on numerous factors. However, one can distinguish between chargeable ticketing systems and open-source software. Regarding Jira and Zendesk, the maximum number of agents that can manage incoming tickets and the functionality range depend on the selected pricing plan. While Jira is free for up to 10 users (but very limited in storage and functionality), Zendesk only offers chargeable plans. Given that 20 agents should use the ticketing system, Jira offers pricing plans starting from 151 euros per month and Zendesk offers pricing plans starting from 500 euros per month, which makes Jira the more affordable system. The capacity of Trac and Znuny is theoretically unlimited, but practically depends on the server's configuration, which may lead to additional costs. Not to mention, in any case, open-source software is the more affordable option.

Concluding, by exploring the five ticketing systems, I was able to make visible the differences and similarities of SDS and to point out the advantages and disadvantages for each of them, hence deriving appropriate use cases. The high integration capability and usability of Zendesk makes it a suitable tool for customer support, where technically less skilled workers have to manage a large number of day-to-day interactions while ensuring customer satisfaction. Software development and project management, on the other hand, is not sufficiently possible with Zendesk, as code integration and collaborative work does not belong to its strengths. In contrast, GitLab can be easily integrated into software development and maintenance, making it an adequate tool for issue tracking of IT services and project management. Znuny and Trac provide clear and most importantly, highly scalable tools for issue tracking that are lightweight and minimalistic and being open-source, they are interesting for small companies in the IT sector that bring technical expertise but a low budget. Finally, Jira's large functionality range makes it a suitable system for every use case, with project management being its biggest strength. Although Jira is not free and a decent training period is needed to get to know the workflow in using Jira, its large configuration opportunities make it interesting for various professional fields beyond ITSM and its relevance in the global market can not be denied.

4 Discussion

The importance of using service desk systems to implement the ITIL guideline is widely accepted [TT13], but the current state of ticketing systems is yet to be improved. Although cost-effectiveness is generally expected when using ticketing systems as an SDS, there are challenges that may hinder their success. Jäntti [Jän12] explored a variety of challenges when setting up a ticketing system and pointed out that wrong classification of tickets is one of the main problems that can occur. This is especially important for ticketing systems in which a user is prompted to assign a category when creating a ticket. The entered category might need clarification or be entirely wrong, leading to additional work to do on the agent's side. Jäntti [Jän12] hence suggests clarifying the meaning of each ticket category clearly for external users, such as customers, or to fully remove the option of categorizing the ticket. Moreover, the author mentioned that the difficulty to identify repeating incident requests as such is another important challenge in the use of SDS. Regarding the migration from an established SDS to a new one, it does always come with a risk and taking too much time to get used to a new workflow can be costly [Yam+19]. Another factor that is often taken into consideration is incorrect ticket assignment, as mentioned by Al-Hawari & Barham [AB21]. When a ticket is assigned to an agent that does not have the time or the technical knowledge to work on it, it has to be reassigned. This significantly increases the time frame until ticket closure, something that has been shown to substantially influence customer satisfaction [IAG16]. Concluding, ticketing systems come with a risk, however, by carefully planning the user's access rights and selecting a suitable system, most of them can be minimized.

Apart from the risks of introducing a ticketing system, there has been extensive research on the factors that are suspected to impede the otherwise efficient workflows in SDS, so-called *bottlenecks* of the system [FDW22; QSS18]. In order to remove these bottlenecks, the use of Artificial Intelligence (AI) and ML is increasingly taken into consideration. The ultimate goal in each case is to automate the ticketing system, such that tasks that were previously manually done are performed automatically. The desired outcome is that more time is left for the human workers to focus on the important customer interactions, thereby increasing the efficiency of the SDS.

Fuchs et al. [FDW22] and Qamili et al. [QSS18] identified ticket assignment as a common bottleneck of SDS, since it is time-consuming and mostly manually done by human workers. Therefore, it may be performed by less-skilled workers or may be completely outsourced. In any case, ticket assignment slows down the entire lifecycle of a ticket and much research is going into automating it. Because it can also be seen as a classification problem, Qamili et al. [QSS18] used Random Forest and Support Vector Machine (SVM) to automatically assign tickets in a predefined scenario and achieved a classification accuracy of 86 percent. Al-Hawari & Barham [AB21] pointed out that machine learning algorithms could also be useful for assisting users to select the correct assignment or category from the start, thereby solving the problem of wrong ticket categorization mentioned by Jäntti [Jän12]. The achievement of more robust classification rates is hence topic of possible future research on automatic ticket assignment.

The authors in [QSS18] also identified working on spam tickets as another bottleneck. Spam tickets are a common issue in SDS environments and manually filtering them out requires time that can be used more efficiently. Therefore, the need for automatic spam filtering is given. In their ML pipeline, the authors were able to reach a precision of 0.97 using a Naive Bayes classificator. However, the false positive rate is especially important at this point, because wrongfully filtered tickets can quickly lead to high customer dissatisfaction. Using an ensemble model, Qamili et al. [QSS18] were able to reduce the false positive rate to a minimum. Nevertheless, a reduction in precision has to be accepted for avoiding false positives.

Sentiment analysis is an important tool to analyze customer satisfaction, but manually carrying out surveys is time-consuming and hence costly [FDW22]. In order to automatically obtain this information, the tickets in a SDS may provide enough material to retrieve the customer's overall mood. This is not only useful for reports on customer satisfaction, but could also be used to automatically assign higher priorities when negative sentiments are recognized in a ticket request [FDW22]. Again, Qamili et al. [QSS18] made use of ML tools to carry out a sentiment analysis, but due to missing ground truth annotations, the authors were not able to provide accuracy measures for their analysis. A similar bottleneck is the prediction of request escalation, as mentioned by Fuchs et al. [FDW22]. When a ticket is escalated, it is brought up to a higher level in the agent's hierarchy. The sooner such a ticket is escalated, the better. Therefore, predicting whether or not a ticket needs to be escalated, may save the system's users time and money. Little research is available for request escalation prediction, but there are some examples such as the study of Montgomery et al. [Mon+18], where the authors were able to predict ticket escalation with 81 percent accuracy. However, it remains a fairly neglected research field.

In conclusion, the automation of ticketing systems is a topic of a wide range of research and is becoming increasingly relevant. Apart from the bottlenecks described above, there is also research going on about automated ticket responses [Sha+19], but the level to which a system should be automated has to be carefully examined. Fuchs et al. [FDW22] reported that customer satisfaction decreased when it became obvious that ticket creators were interacting with a chatbot instead of a human support agent. Nevertheless, bottlenecks in ticketing systems impede the efficiency of modern SDS and recent advances in ML lead to many possibilities to integrate AI into process automation in ticketing systems, with the ultimate goal of saving time needed to perform repetitive or unnecessary tasks. The features that are useful for training AI models in this context are yet to be explored, but Al-Hawari & Barham [AB21] found that the interaction taking place after ticket creation (e.g. in a commenting function) can be used to significantly improve classification accuracies. However, future research is needed to examine further aspects that drive ticketing systems to higher automation levels.

5 Conclusion

Ticketing systems have become an indispensable tool for efficient IT Service Management (ITSM) by serving as a single point of contact for customer interaction, task management and issue tracking. However, as the amount of ticketing systems on the market and their corresponding customization options are continuously increasing, the decision for or against a specific system setup is a challenging one. In this term paper, I provided an overview about the usages and benefits of ticketing systems and reported hands-on experience on five different systems. Depending on the use case, a ticketing system may be more or less suitable. Moreover, the available budget, the technical expertise of the users and the expected amount of incoming tickets play an important role to make a sound decision. In this comparison, I found Jira and Zendesk to be especially user-friendly, however, some training period is needed and the two systems come with a cost. On the other hand, GitLab is a widely recognized issue tracker and its ticketing system is sufficient to handle even comprehensive software development. Lastly, Znuny and Trac are open-source ticketing systems offering less usability, especially for non-experts, but provide reliable tools for issue tracking and importantly, are freely scalable. Regarding future research, a lot of effort within the last years went into exploring the integration of Machine Learning (ML) into ticketing systems to automate time-consuming processes. While some of these attempts turned out to successfully improve the efficiency in day-to-day tasks, the automation of ticketing systems continues to be an ongoing challenge and remains relevant for moving away from traditional service desks towards modern solutions in ITSM.

References

[AB21] Feras Al-Hawari and Hala Barham. “A machine learning based help desk system for IT service management”. In: *Journal of King Saud University-Computer and Information Sciences* 33.6 (2021), pp. 702–718. DOI: https://doi.org/10.1016/j.jksuci.2019.04.001.

[Agl+22] Kent Darryl Aglibar et al. “Ticketing system: A descriptive research on the use of ticketing system for project management and issue tracking in it companies”. In: *International Journal of Computing Sciences Research* (2022), pp. 1–10. DOI: https://doi.org/10.48550/arXiv.2202.06213.

[Cho+20] Prithwiraj Choudhury et al. “GitLab: work where you want, when you want”. In: *Journal of Organization Design* 9 (2020), pp. 1–17. DOI: https://doi.org/10.1186/s41469-020-00087-8.

[Eck10] Wayne W Eckerson. *Performance dashboards: measuring, monitoring, and managing your business*. John Wiley & Sons, 2010.

[FDW22] Simon Fuchs, Clemens Drieschner, and Holger Wittges. “Improving support ticket systems using machine learning: A literature review”. In: *Proceedings of the 55th Hawaii International Conference on System Sciences* (2022), pp. 1893–1902. URL: http://hdl.handle.net/10125/79570.

[Foo+00] Schubert Foo et al. “An integrated help desk support for customer services over the World Wide Web—a case study”. In: *Computers in Industry* 41.2 (2000), pp. 129–145. DOI: https://doi.org/10.1016/S0166-3615(99)00037-8.

[GK19] Florika Gohil and M Vikash Kumar. “Ticketing system”. In: *International Journal of Trend in Scientific Research and Development* 3.4 (2019), pp. 155–156.

[IAG16] Roumiana Ilieva, Kiril Anguelov, and Delyana Gashurova. “Monitoring and optimization of e-Services in IT Service Desk Systems”. In: *2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA)*. IEEE, 2016, pp. 1–4. DOI: https://doi.org/10.1109/SIELA.2016.7543013.

[Jän12] Marko Jäntti. “Examining challenges in IT service desk system and processes: a case study”. In: *The Seventh International Conference on Systems (ICONS)*. 2012, pp. 105–108.

[MLM22] Lloyd Montgomery, Clara Lüders, and Walid Maalej. “An alternative issue tracking dataset of public jira repositories”. In: *Proceedings of the 19th International Conference on Mining Software Repositories*. 2022, pp. 73–77. DOI: https://doi.org/10.1145/3524842.3528486.

[Mon+18] Lloyd Montgomery et al. “Customer support ticket escalation prediction using feature engineering”. In: *Requirements Engineering* 23 (2018), pp. 333–355. DOI: https://doi.org/10.1007/s00766-018-0292-3.

[PM90] Jeffrey K Pinto and Samuel J Mantel. “The causes of project failure”. In: *IEEE transactions on engineering management* 37.4 (1990), pp. 269–276. DOI: https://doi.org/10.1109/17.62322.

[QSS18] Ruanda Qamili, Shaban Shabani, and Johannes Schneider. "An intelligent framework for issue ticketing system based on machine learning". In: *2018 IEEE 22nd International Enterprise Distributed Object Computing Workshop (EDOCW)*. IEEE, 2018, pp. 79–86. DOI: `https://doi.org/10.1109/EDOCW.2018.00022`.

[RKK18] Eka Rachmawati, M Kom, and M Kom. "Web-Based Ticketing System Helpdesk Application Using CodeIgniter Framework (Case Study: PT Commonwealth Life)". In: *International Journal of Computer Science and Mobile Computing* 7.12 (2018), pp. 29–41.

[Sha+19] Kuruparan Shanmugalingam et al. "Corporate it-support help-desk process hybrid-automation solution with machine learning approach". In: *2019 Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2019, pp. 1–7. DOI: `https://doi.org/10.1109/DICTA47822.2019.8946083`.

[TM16] Anel Tanovic and Nikos Mastorakis. "Advantage of using Service Desk Management Systems in real organizations". In: *International Journal of Economics and Management Systems* 1 (2016), pp. 81–86.

[TT13] Xiaojun Tang and Yuki Todo. "A study of service desk setup in implementing IT service management in enterprises". In: *Technology and Investment* (2013), pp. 190–196. DOI: `http://dx.doi.org/10.4236/ti.2013.43022`.

[Yam+19] Hiromi Yamaoka et al. "Case study of implementing an it service desk ticketing system at small computer center". In: *Proceedings of the 2019 ACM SIGUCCS Annual Conference*. Association for Computing Machinery, 2019, pp. 140–144. DOI: `https://doi.org/10.1145/3347709.3347820`.

[ZGG09] Chen Zhao, HongHua Gan, and Fei Gao. "A study on the process model for IT service management". In: *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering*. 3. World Scientific, Engineering Academy, and Society, 2009, pp. 206–210.