

IHRE SCHULUNG BEI ATLASTEAM

# EazyBI Training - Business Intelligence Reports Für Jira Und Confluence



**ATLASTEAM**

Ihr Berliner Atlassian Partner

[www.atlasteam.de](http://www.atlasteam.de)

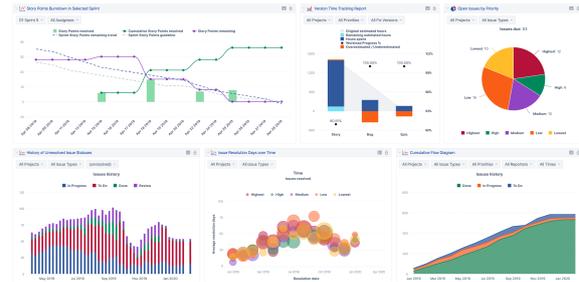
**Sprache:** Deutsch, Englisch auf Wunsch

**Level 3 - Profi:** Erfahrungen werden empfohlen

**Dauer:** 6 Stunden

**Kosten:** 1.920,00 € netto

- Schulungsinstanz DC/Cloud
- max. 3 Teilnehmer\*innen
- Für **Data Manager / Data Scientists**
- Für **Portfolio-** oder **C-Level Manager**, mit Kenntnissen in **Business Intelligence**
- Für **Data Driven Project Managers (BI Skills)** sind zwingend erforderlich)
- Übungen
- Support nach der Schulung auf T&M Basis möglich
- Preisnachlässe für weitere Teilnehmer aus einem Unternehmen



Für Data Manager / Data Scientists, Portfolio- oder C-Level Manager

## Themen

- Individuelle **Auswertungen** zu mehreren **Projekten** aus Jira
- Individuelle **Auswertungen** zu mehreren **Bereichen** aus Confluence
- **Einbindung** von einzelnen Reports aus Jira **auf Confluence Seiten**
- **Extract-Transform-Load (ETL)** in das eazyBI **DWH** mit JavaScript beim regelmäßigen Aktualisieren der Daten aus Jira/Confluence
- Anpassung der Daten im eazyBI DWH je Report mit **MDX**
- Automatische **Benachrichtigungen**, wenn Kennzahlen in einem Report unter- oder überschritten werden

## Beispiele für Übungen

- Mitarbeiter\*in des Monats, anhand selbst definierter Kennzahlen aus Jira oder Confluence
- Kosten je Mitarbeiter\*in via Excel einlesen und mit den gebuchten Zeiten multiplizieren. Als Ergebnis werden bspw. die Kosten je Release angezeigt.
- Kosten je Mitarbeiter\*in via Excel einlesen und mit den geschätzten Zeiten multiplizieren. Als Ergebnis werden bspw. die Kosten der noch anstehenden Arbeit je Release angezeigt.
- Geleistete Kosten und anstehende Kosten gegen Projekt-Budgets rechnen und darstellen.
- Zusammenfassung von Werten mit MDX wie "Highest, High, Medium" zu "high" und "low", "Lowest" zu "low". Ein grundsätzliches Verständnis

- Einbindung **externer Daten**, wie Microsoft Excel und die kombinierte Darstellung dieser Daten mit Daten aus Jira
- Einbindung externer Daten via **SQL** und **REST**
- Optische Anpassung von Reports, eazyBI bietet viele Varianten
- Externe Generierung von Daten mit **Python** und **Pandas** und das Einlesen dieser Daten in ein eazyBI DWH

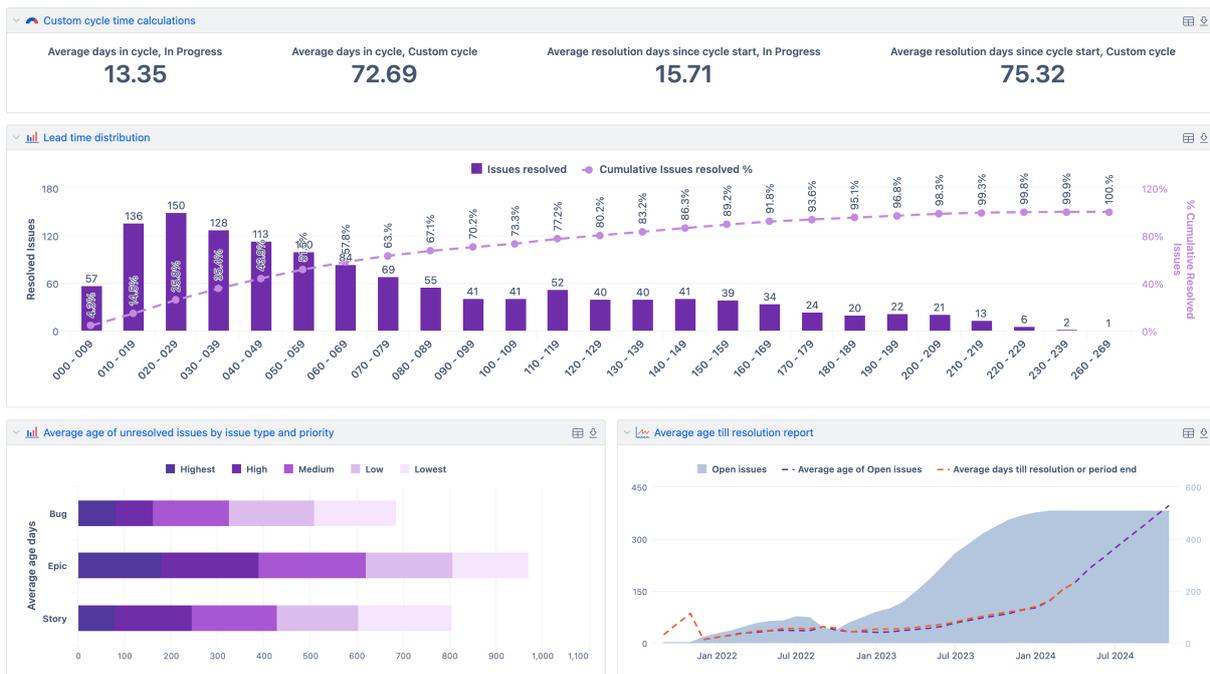
hilft, wenn bspw. "gut" und "schlecht" anteilig je einer Zeitspanne angezeigt werden soll.

- Erzeugung eines Time-Sheets.
- Darstellung von Projektplänen als Gantt.
- Weitere Beispiele. **Es können auch Wünsche geäußert werden.**

## Beispiele für Reports

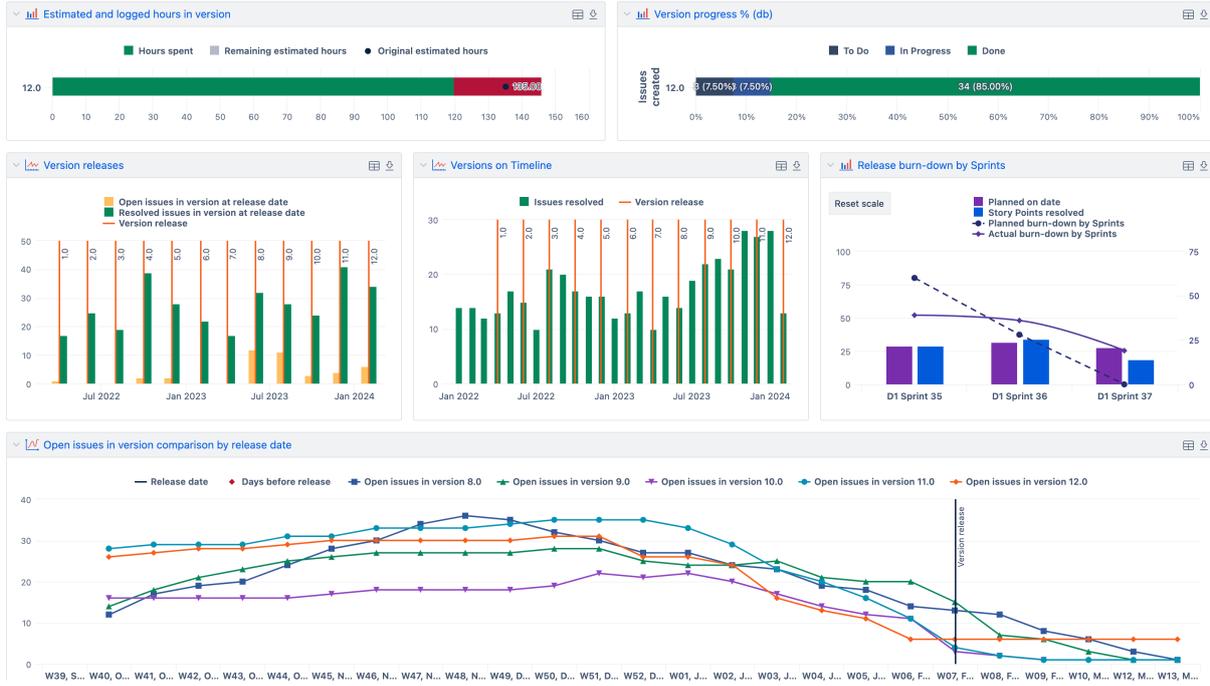
Welche gemeinsam von Beginn an individuell und gemeinsam erzeugt werden.

Wie viele Aufgaben sind offen, verteilt über Zeiträume, wie lange Aufgaben bereits offen sind

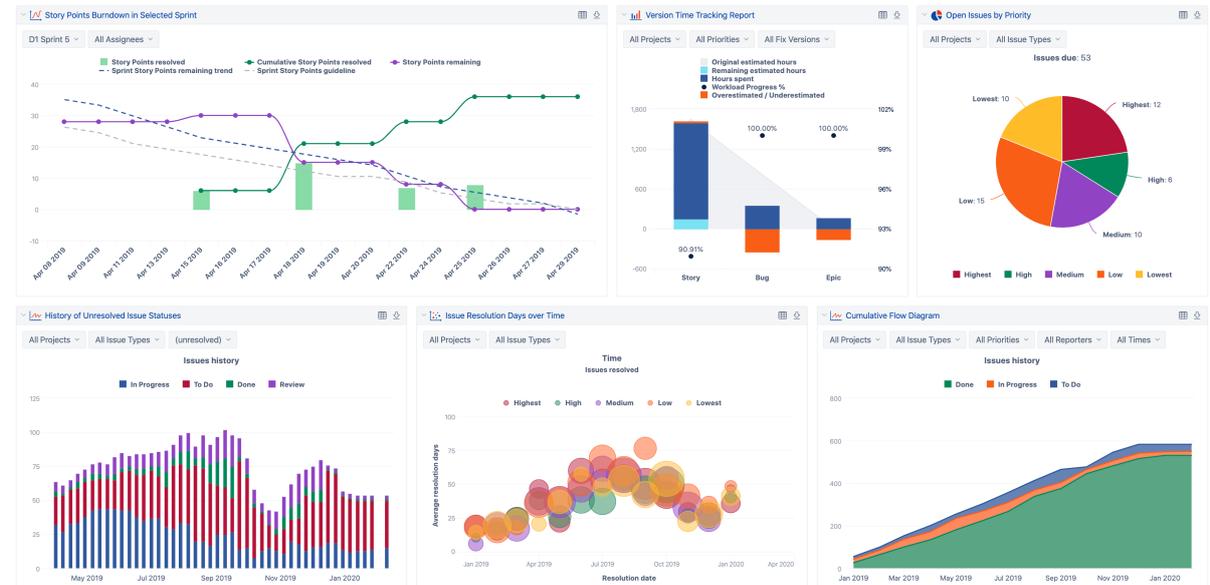


An welchen Releases wird gearbeitet, nach Fertigstellungsgrad

Hier ist eine **Wasserfall-Planung** möglich, mit einer agilen Darstellungsform, wie **Burn-Down Charts**.



## Wie entwickelt sich die Arbeitsumfang über die Zeit



# Viel-dimensionale Tabellen

Show available dimensions

**Pages** Nonempty

- > Project
- > Issue Type
- > Priority
- > Assignee ↕ ✕

**Rows** Nonempty

- > Project
- > Issue Type

**Columns** > Measures > Priority

Table Bar Line Pie Scatter Timeline Map Gantt Gauge

🔗 ↺ ↻ 🗨 📄 Hide empty Total

DEMO Alpha, DEMO Beta ▼ Standard ▼ All Priorities ▼ All Assignees ▼

		Average age days		
		Sum	Low	Lowest
+ DEMO Alpha	Bug	13.37	192.57	189.36
	Epic		201.49	257.47
	Story	35.16	173.58	202.91
+ DEMO Beta	Bug	76.28	175.34	173.72
	Epic	24.17	187.73	158.07
	Story	30.57	163.38	216.58

- 🔗 Drill across
- Select this
- 🔖 Bookmark
- 🗑 Remove
- ↕ Order by
- ↓ Top rows
- ↑ Bottom rows
- 🔍 Filter rows
- 🔢 Cell formatting

# Berechnung der Zukunft

All Projects

CalcM Open Issues

Category	Peak (Approx. Date)	Peak Value (Approx.)	Current Value (Mar 2024)
All Assignees	Nov 2021	100	40
Individual Users	Various (2021-2022)	30	10-20

Seite 5 von 9

# Over-Allocation And Duration

Kanban A In Progress 2022



## Programmierung

### MDX

**Define calculated member formula**

[Measures].[ Status last update 8-14 days ago ] =

```

1 NonZero(Count(
2   Filter(
3     Descendants([Issue].CurrentMember, [Issue].[Issue]),
4     -- for unresolved issues only
5     isEmpty([Measures].[Issue resolution date]) and
6     -- set interval
7     DateDiffDays(
8       [Measures].[Issue status updated date],
9       Now()
10      ) > 7 and
11     DateDiffDays(
12       [Measures].[Issue status updated date],
13       Now()
14      ) <= 14 and
15     [Measures].[Issues cr] >0
16   )
17 ))

```

Formatting #,### Integer

Read calculated members tutorial to learn about calculated member formulas. You can select members, operators and frequently used functions from sidebar to insert them into calculated member formula.

Predicted dates Version report Start Date Time in version

Cumulative Story Points resolved till today

Version burn-down Version burn-up

Overestimated / Underestimated

Versions released in period Unreleased versions in period

**Other dimensions**

Priority Time Project Issue Type

**Operators**

+ - \* / ( ) and or not

< <= = >= > <>

**Member functions**

.CurrentMember .PrevMember ParallelPeriod

**Set functions**

.Members .Children Descendants Filter

**Aggregators**

Count Sum Avg Median Aggregate

Update
Delete
cancel

# JavaScript

## eazyBI settings

General License Additional Advanced settings

### Advanced settings

Please see the advanced settings help page.

```
1 [jira]
2 common_javascript_code = ''
3 // may contain several functions
4
5
6 // The function setNumber validates and converts string values to numeric values
7 function setNumber(str)
8 {
9     num = Number(str);
10    if ( Object.prototype.toString.call(num) === "[object Number]" && !isNaN( num ) ) {
11        return num;
12    }
13    else {
14        return null;
15    }
16 }
17
18
19 // The function workdays calculates workdays from two dates
20 function workdays ( from, to ) {
21     var c1 = new Date(Date.parse(from));
22     var c2 = new Date(Date.parse(to));
23     var days = 0.0;
24     var fullWeeks = 0;
25     var nonworkCount = 0;
26     nonworkdays = [0,6]; // non-working days Saturday and Sunday
27
28     // check that from date is before to date
29     if (c1 > c2) return -workdays(to, from );
```

## Python Pandas

```
df['Story'] = (args.count * df['Story Factor'] / df['Last Day']).astype(dtype='int64')
df['Bug'] = (args.count * df['Bug Factor'] / df['Last Day']).astype(dtype='int64')
df = df.melt(id_vars=cols, var_name='issueType', value_name='Issue Type Count', ignore_index=False)
df = df.drop(columns=['Story Factor', 'Bug Factor'])

# Create a row per issue.
# No need to take care of issue types since that piece of information got melted before, and now
# We only get the index with dates and issues types, and the column with the amount of issues for
df = df.reset_index()
df = df.set_index(['created', 'issueType']).sort_index(level=0)
_idx = df.index.values.tolist()
_icount = df['Issue Type Count'].to_list()
# We duplicate all tuples (original index) for the amount of issues for that day. One tuple represents
_idx_new = [_idx[i] for i in range(0, len(_idx)) for j in range(0, _icount[i])]
# This is our issue counter. KA-1, KA-2, and so on.
_icount_new = [i for i in range(1, len(_idx_new)+1)]
_df_issues = pd.DataFrame({'Issue Counter':_icount_new}, index=pd.MultiIndex.from_tuples(_idx_new))
# By merging both results by inner we drop all unrelated days
df = df.merge(_df_issues, how='inner', on=['created', 'issueType']).ffill()
df = df.drop(columns=['Issue Type Count'])

# Add issue fields
df = df.reset_index()
df['summary'] = df['key'] + '-' + df['Issue Counter'].astype(dtype='string') + ' ' + df['issueType']
df['reporter'] = reporter
df['priority'] = 'Medium'
_estimates_min_max_mean = (estimated_hrs_max-estimated_hrs_min)/2+estimated_hrs_min
# For estimates we apply a gaussian curve to get a more realistic distribution. 0.8 shifts mean
_estimates_floats = truncnorm((estimated_hrs_min-_estimates_min_max_mean*0.8)/4, (estimated_hrs_max-_estimates_min_max_mean*0.8)/4, (estimated_hrs_min-_estimates_min_max_mean*0.8)/4, (estimated_hrs_max-_estimates_min_max_mean*0.8)/4)
# A division by four and round multiplied by four creates a distance of four between all values.
df['originalEstimate'] = list(round(i/4)*4 for i in _estimates_floats)
```

Haben Sie weitere Fragen, besuchen Sie unsere [FAQ-Seite](#) oder [kontaktieren](#) Sie uns.

## Haben Sie noch Fragen?

Gerne beantworten wir Ihre Fragen oder unterstützen Sie bei der Buchung:

E-Mail: [team@atlasteam.de](mailto:team@atlasteam.de)

Telefon: +49 (0)30 9599840-20

Web: [www.atlasteam.de](http://www.atlasteam.de)

Übersicht über alle Schulungen: [www.atlasteam.de/schulungen-uebersicht](http://www.atlasteam.de/schulungen-uebersicht)

## Kennen Sie schon diese Produkte & Leistungen?

### Offsite-Backup Ihrer Atlassian-Cloud-Instanzen

Wir unterstützen Sie bei der verlässlichen Offsite-Sicherung Ihrer Daten aus der Cloud und von Data-Center. Ihre Daten werden ausschließlich in Deutschland und von deutschen Unternehmen verarbeitet und verschlüsselt gesichert.

Mehr unter:

[www.atlasteam.de/services/jira-confluence-atlassian-cloud-backup](http://www.atlasteam.de/services/jira-confluence-atlassian-cloud-backup)



### €-Rechnung und Rabatte für Ihre Lizenzen

Mehr unter: [www.atlasteam.de/lizenzen-und-hosting/jira-confluence-atlassian](http://www.atlasteam.de/lizenzen-und-hosting/jira-confluence-atlassian)