

# Agile Metriken

## Qualitätssicherung in agilen Teams

### Zielsetzung

„Scrum ist eine Implementierung der Theorie des Empirismus, der zufolge Wissen auf Erfahrung beruht und Entscheidungen auf der Grundlage dieses Wissens getroffen werden. Dazu sind drei Voraussetzungen zu schaffen:

1. Transparenz
2. Überprüfung
3. Anpassung“<sup>1</sup>

Ziel dieser Arbeit war es, Metriken zu ermitteln, die Qualitätsprobleme im Entwicklungsprozess oder im Softwareprodukt quantitativ abbilden können und somit der Forderung nach Überprüfung gerecht werden.

### Vorgehensweise

Es wurde eine Fallstudie wie in „Agile Metrics In Action“<sup>2</sup> beschrieben mit einem Scrum Team durchgeführt und dabei in folgenden Schritten vorgegangen:

1. Metriken ermitteln  
Die Metriken wurden über die sogenannte GQM (Goal-Question-Metric)-Methode, bei der die Daten der letzten Sprint Retrospektiven ausgewertet wurden, und ergänzend über eine Umfrage im Team ermittelt.
2. Software erstellen  
Die im ersten Schritt ermittelten Metriken wurden über eine eigens erstellte Software ermittelt, aufbereitet und in einem Elastic Stack gespeichert und dargestellt.
3. Ergebnisse evaluieren  
Das Scrum Team nutzte diese Metriken über einen Zeitraum von nicht ganz drei Sprints in den Retrospektiven, um eine bessere Einsicht in den Entwicklungsprozess zu bekommen.

### Umsetzung

Es wurde eine Java-Software erstellt, die eigenständig betrieben werden kann und Daten von den einzelnen Systemen im Entwicklungsprozess abholen, aufbereiten und als Metriken in einer Datenbank speichern kann. In der Fallstudie waren die Systeme Jira, Bitbucket, SonarQube und zur Speicherung und Visualisierung diente ein Elastic Stack.

### Quellen

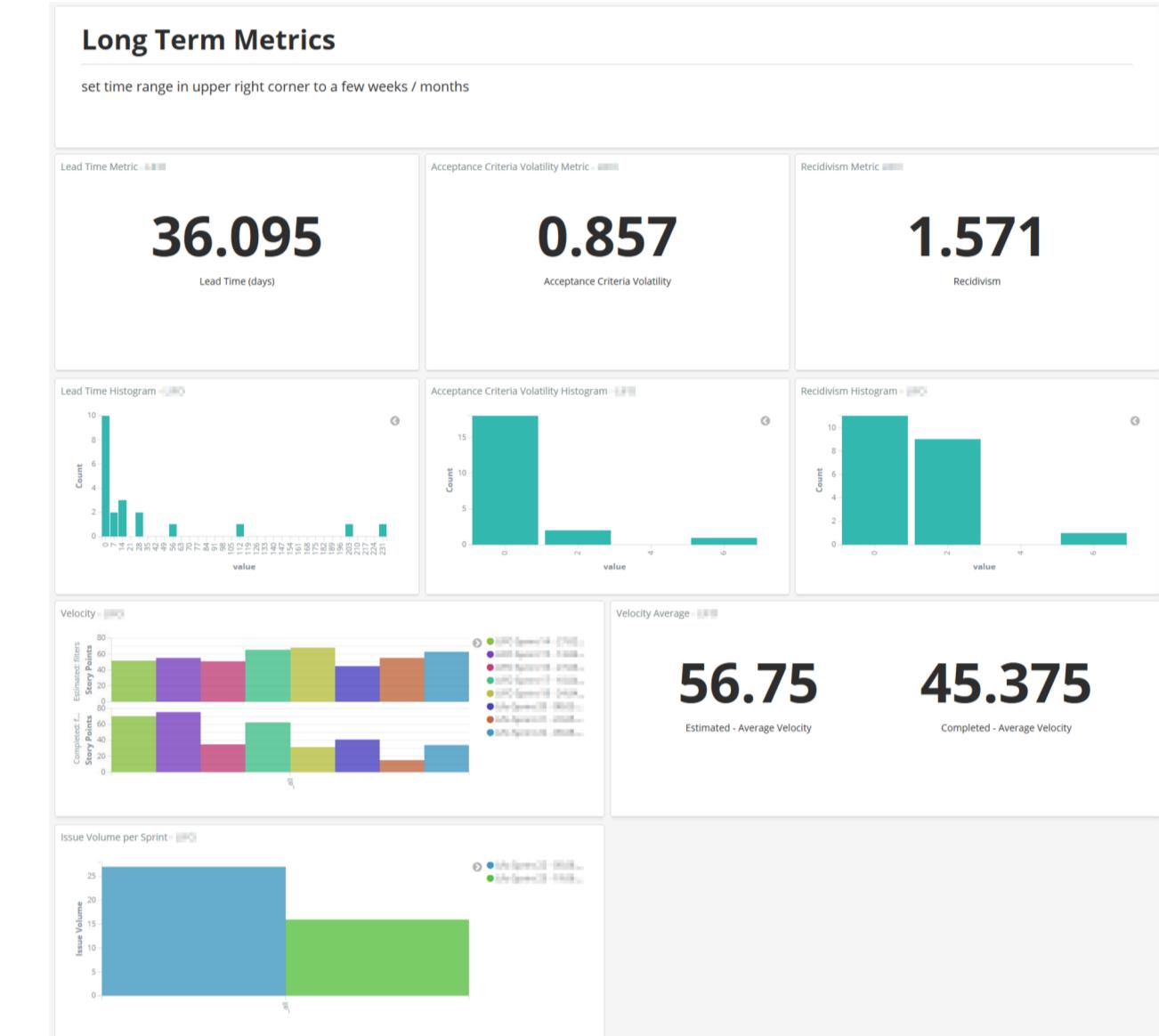
<sup>1</sup> vgl. Rolf Dräther, Holger Koschek und Carsten Sahling. Scrum: kurz & gut. 1. Auflage. O'Reillys Taschenbibliothek. Beijing Cambridge Farnham Köln Sebastopol, Tokyo: O'Reilly, 2013. isbn: 978- 3-86899-833-7, S.14ff

<sup>2</sup> Christopher W. H. Davis. Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams. 1st. Greenwich, CT, USA: Manning Publications Co., 2015. isbn: 978-1-61729-248-4

### Evaluierung

Zur **quantitativen Evaluierung** wurden die Metriken beobachtet. Es waren nach diesen drei Sprints bereits Tendenzen erkennbar, ob diese durch das Dashboard eingetreten sind, war in diesem kurzen Zeitraum nicht eindeutig feststellbar. Was aber klar erkennbar war, war die Wahl der Metriken, da durch diese viele Schwachstellen sichtbar gemacht werden konnten.

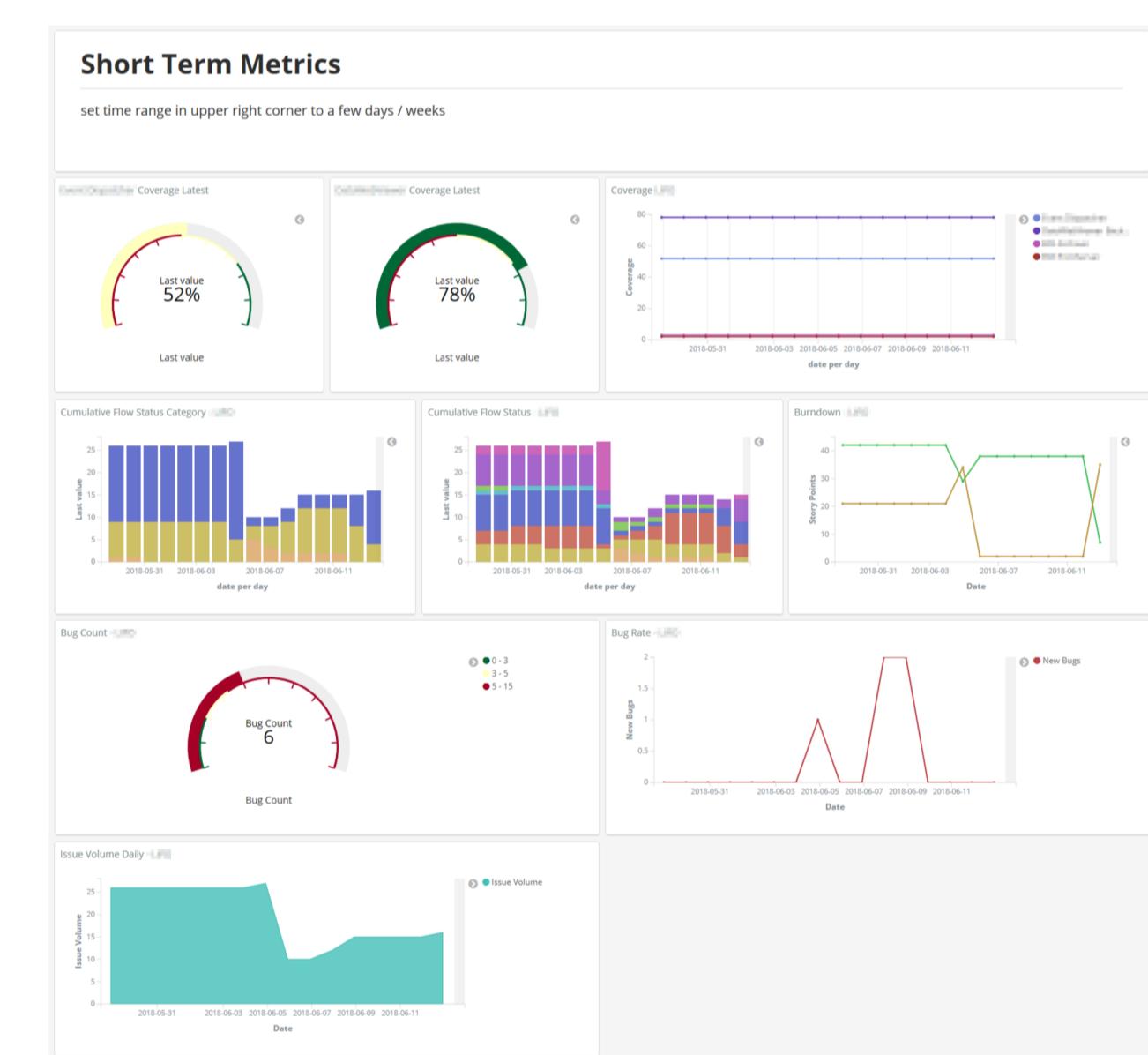
Ergänzend wurde eine **qualitative Evaluierung** in Form von Interviews durchgeführt. Interview wurden dabei eine Entwicklerin, der Product Owner und der Scrum Master. Auffallend dabei war, dass jeder der drei das Dashboard bereits für seine Zwecke zu nutzen wusste. Während der Product-Owner das Dashboard mehr für Werbezwecke genutzt hat, um andere Abteilungen auf die Möglichkeit von Metriken aufmerksam zu machen, hat es der Scrum-Master eher für die Langzeit-Sicht des Teams genutzt und seinen Fokus auf die Verbesserung des Prozesses gelegt. Die Entwicklerin wiederum hatte den Fokus auf die kurzfristigen Metriken, wie den Bug Count, um schnell auf Probleme reagieren zu können.



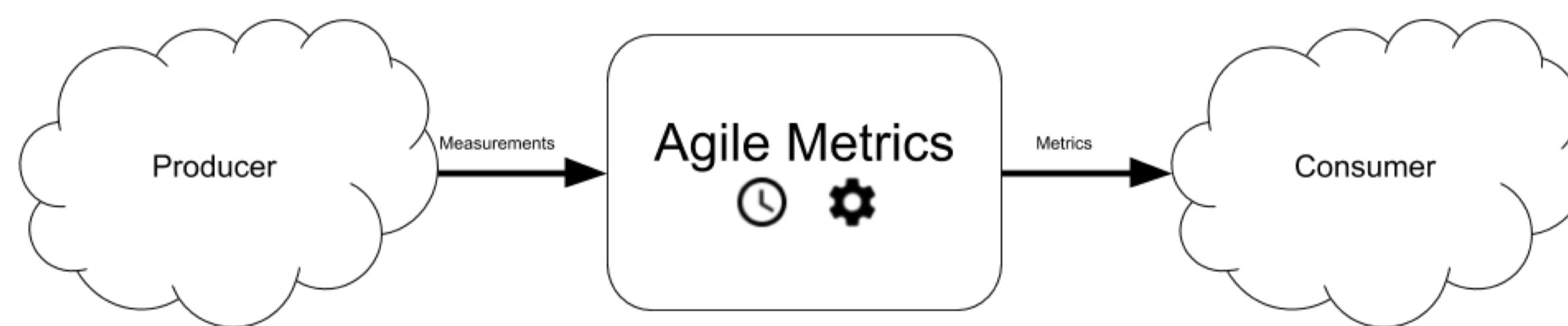
Langzeit-Metriken auf dem Team-Dashboard in Kibana

### Schlussfolgerungen

In dieser Arbeit konnte in einer Fallstudie gezeigt werden, dass mithilfe der GQM-Methodik, ergänzt durch eine Umfrage im entsprechenden Scrum-Team, Metriken ermittelt werden können, die es ermöglichen, die Schwachstellen in einem Produkt und im agilen Prozess in Zahlen zu fassen. Die entwickelte Software hilft dabei, die Daten aus den unterschiedlichen Systemen im Entwicklungsprozess als Metriken aufzubereiten und zu speichern. Für einen einfachen Zugang und eine uneingeschränkte Erweiterbarkeit wurde der Quellcode der Software unter der quelloffenen MIT-Lizenz veröffentlicht. Bei der Visualisierung von Metriken bietet Kibana eine geeignete Plattform, um aus den gespeicherten Metriken einfache Dashboards mit unterschiedlichen Visualisierungen bereitzustellen. Durch den Einsatz der entwickelten Software und der vorgestellten Modelle zur Identifizierung von relevanten Metriken, kann die Qualität in einem agilen Team dadurch erhöht werden, dass Qualitätsprobleme durch Metriken sichtbar gemacht und in den Retrospektiven Gegenmaßnahmen dafür getroffen werden können.



Kurzzeit-Metriken auf dem Team-Dashboard in Kibana



Die entwickelte Software sammelt Metriken in den Systemen entlang des Entwicklungsprozesses und speichert diese aufbereitet zur Visualisierung ab.

# Agile Metriken

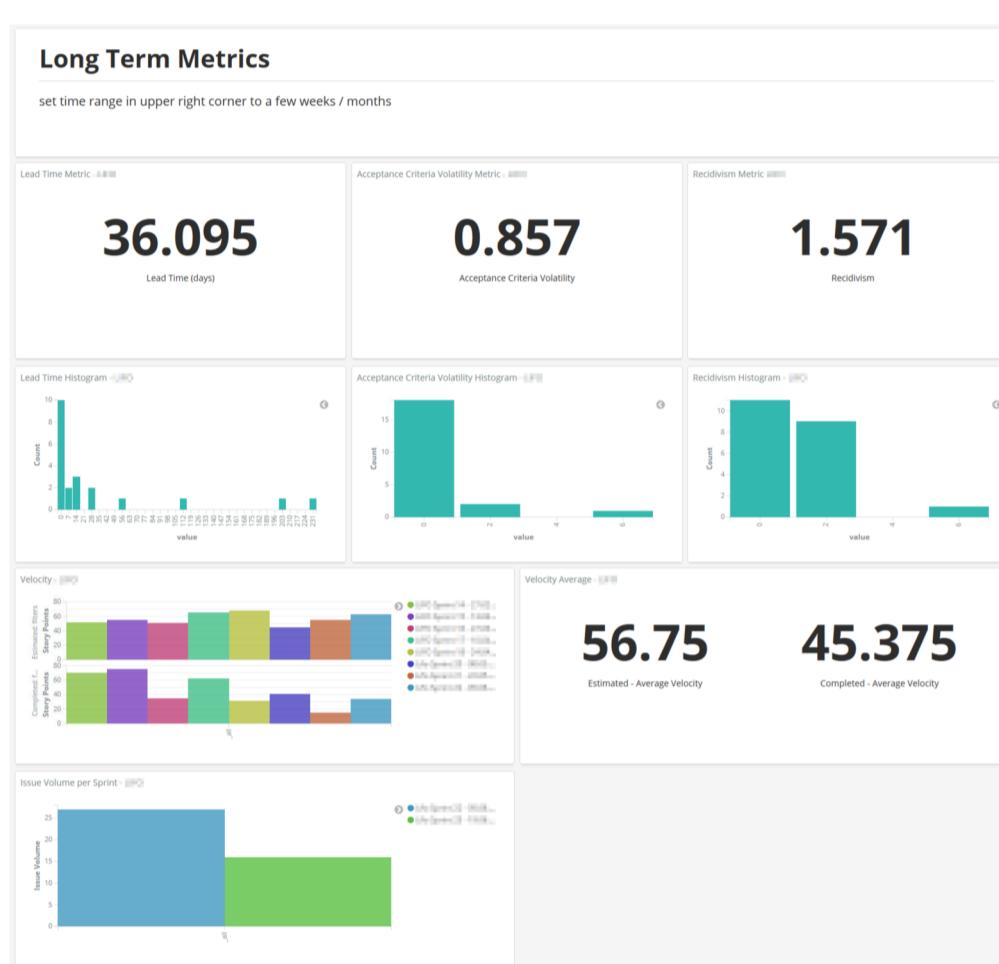
## Qualitätssicherung in agilen Teams

### Zielsetzung

„Scrum ist eine Implementierung der Theorie des Empirismus, der zufolge Wissen auf Erfahrung beruht und Entscheidungen auf der Grundlage dieses Wissens getroffen werden. Dazu sind drei Voraussetzungen zu schaffen:

1. Transparenz
2. Überprüfung
3. Anpassung“<sup>1</sup>

Ziel dieser Arbeit war es, Metriken zu ermitteln, die Qualitätsprobleme im Entwicklungsprozess oder im Softwareprodukt quantitativ abbilden können und somit der Forderung nach Überprüfung gerecht werden.

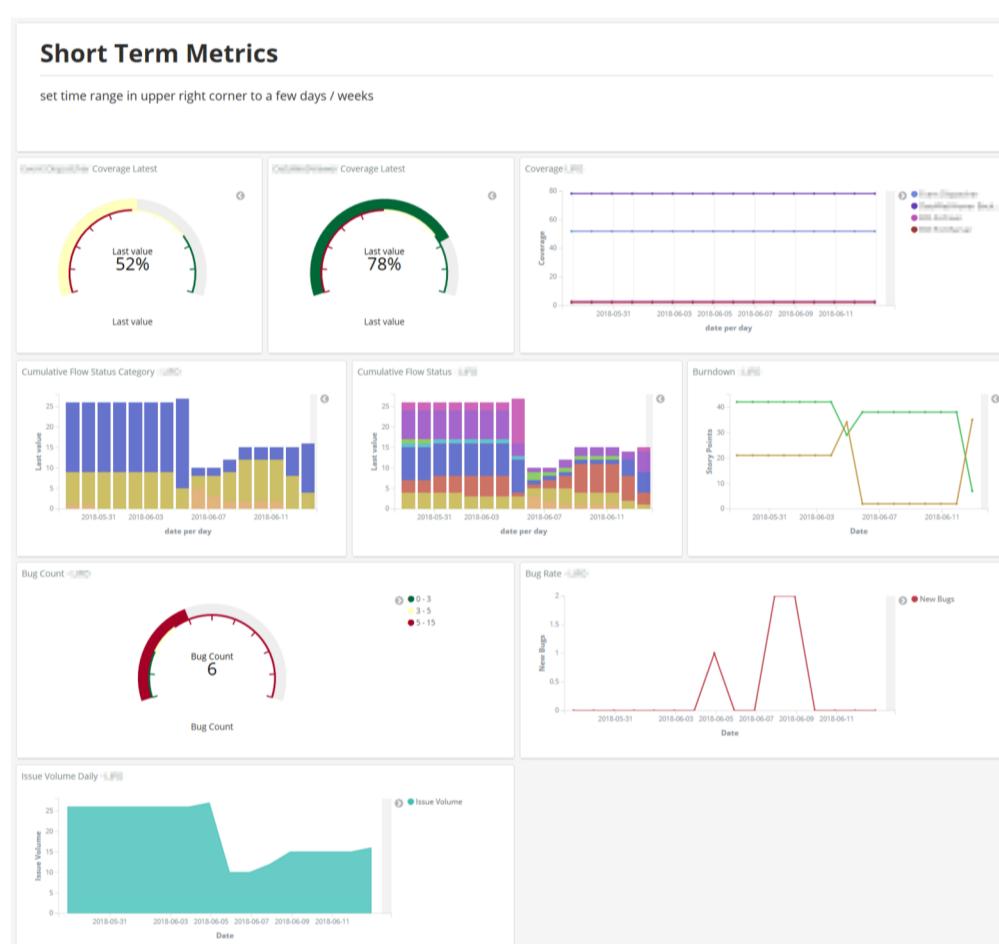


Langzeit-Metriken auf dem Team-Dashboard in Kibana

### Vorgehensweise

Es wurde eine Fallstudie wie in „Agile Metrics In Action“<sup>2</sup> beschrieben mit einem Scrum Team durchgeführt und dabei in folgenden Schritten vorgegangen:

1. Metriken ermitteln  
Die Metriken wurden über die sogenannte GQM (Goal-Question-Metric)-Methode, bei der die Daten der letzten Sprint Retrospektiven ausgewertet wurden, und ergänzend über eine Umfrage im Team ermittelt.
2. Software erstellen  
Die im ersten Schritt ermittelten Metriken wurden über eine eigens erstellte Software ermittelt, aufbereitet und in einem Elastic Stack gespeichert und dargestellt.
3. Ergebnisse evaluieren  
Das Scrum Team nutzte diese Metriken über einen Zeitraum von nicht ganz drei Sprints in den Retrospektiven, um eine bessere Einsicht in den Entwicklungsprozess zu bekommen.



Kurzzeit-Metriken auf dem Team-Dashboard in Kibana

### Evaluierung

Zur quantitativen Evaluierung wurden die Metriken beobachtet. Es waren nach diesen drei Sprints bereits Tendenzen erkennbar, ob diese durch das Dashboard eingetreten sind, war in diesem kurzen Zeitraum nicht eindeutig feststellbar. Was aber klar erkennbar war, war die Wahl der Metriken, da durch diese viele Schwachstellen sichtbar gemacht werden konnten.

Ergänzend wurde eine qualitative Evaluierung in Form von Interviews durchgeführt. Interviewt wurden dabei eine Entwicklerin, der Product Owner und der Scrum Master. Auffallend dabei war, dass jeder der drei das Dashboard bereits für seine Zwecke zu nutzen wusste. Während der Product-Owner das Dashboard mehr für Werbezwecke genutzt hat, um andere Abteilungen auf die Möglichkeit von Metriken aufmerksam zu machen, hat es der Scrum-Master eher für die Langzeit-Sicht des Teams genutzt und seinen Fokus auf die Verbesserung des Prozesses gelegt. Die Entwicklerin wiederum hatte den Fokus auf die kurzfristigen Metriken, wie den Bug Count, um schnell auf Probleme reagieren zu können.

### Schlussfolgerungen

In dieser Arbeit konnte in einer Fallstudie gezeigt werden, dass mithilfe der GQM-Methodik, ergänzt durch eine Umfrage im entsprechenden Scrum-Team, Metriken ermittelt werden können, die es ermöglichen, die Schwachstellen in einem Produkt und im agilen Prozess in Zahlen zu fassen. Die entwickelte Software hilft dabei, die Daten aus den unterschiedlichen Systemen im Entwicklungsprozess als Metriken aufzubereiten und zu speichern. Für einen einfachen Zugang und eine uneingeschränkte Erweiterbarkeit wurde der Quellcode der Software unter der quellöffentlichen MIT-Lizenz veröffentlicht. Bei der Visualisierung von Metriken bietet Kibana eine geeignete Plattform, um aus den gespeicherten Metriken einfache Dashboards mit unterschiedlichen Visualisierungen bereitzustellen. Durch den Einsatz der entwickelten Software und der vorgestellten Modelle zur Identifizierung von relevanten Metriken, kann die Qualität in einem agilen Team dadurch erhöht werden, dass Qualitätsprobleme durch Metriken sichtbar gemacht und in den Retrospektiven Gegenmaßnahmen dafür getroffen werden können.

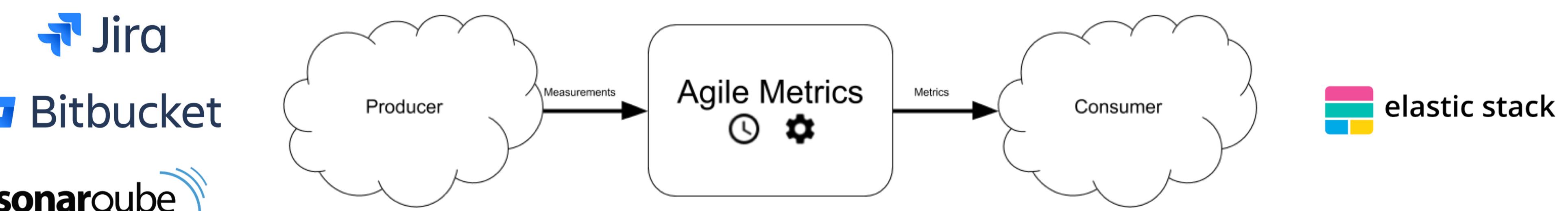
### Umsetzung

Es wurde eine Java-Software erstellt, die eigenständig betrieben werden kann und Daten von den einzelnen Systemen im Entwicklungsprozess abholen, aufbereiten und als Metriken in einer Datenbank speichern kann. In der Fallstudie waren die Systeme Jira, Bitbucket, SonarQube und zur Speicherung und Visualisierung diente ein Elastic Stack.

### Quellen

<sup>1</sup> vgl. Rolf Dräther, Holger Koschek und Carsten Sahling. Scrum: kurz & gut. 1. Auflage. O'Reillys Taschenbibliothek. Beijing Cambridge Farnham Köln Sebastopol, Tokyo: O'Reilly, 2013. isbn: 978-3-86899-833-7, S.14ff

<sup>2</sup> Christopher W. H. Davis. Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams. 1st. Greenwich, CT, USA: Manning Publications Co., 2015. isbn: 978-1-61729-248-4

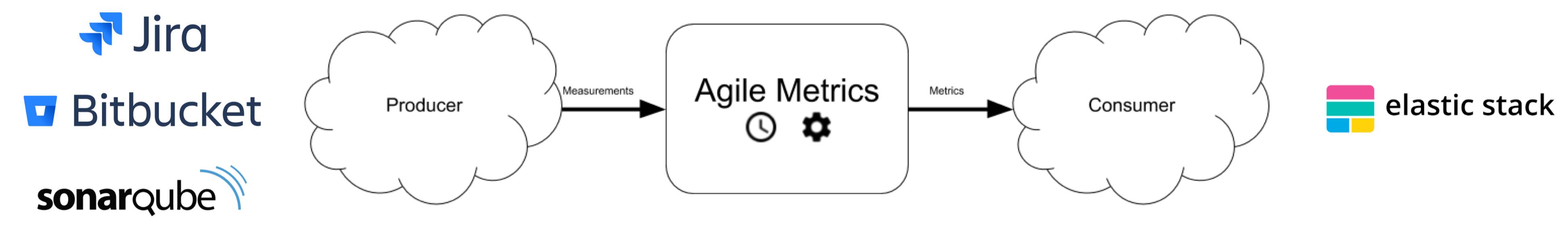


Die entwickelte Software sammelt Metriken in den Systemen entlang des Entwicklungsprozesses und speichert diese aufbereitet zur Visualisierung ab.

# Agile Metriken

## Qualitätssicherung in agilen Teams

Zielsetzung	Evaluierung
<p>„Scrum ist eine Implementierung der Theorie des Empirismus, der zufolge Wissen auf Erfahrung beruht und Entscheidungen auf der Grundlage dieses Wissens getroffen werden. Dazu sind drei Voraussetzungen zu schaffen:</p> <ol style="list-style-type: none"> <li>1. Transparenz</li> <li>2. Überprüfung</li> <li>3. Anpassung“<sup>1</sup></li> </ol> <p>Ziel dieser Arbeit war es, Metriken zu ermitteln, die Qualitätsprobleme im Entwicklungsprozess oder im Softwareprodukt quantitativ abbilden können und somit der Forderung nach Überprüfung gerecht werden.</p>	<p>Zur <b>quantitativen Evaluierung</b> wurden die Metriken beobachtet. Es waren nach diesen drei Sprints bereits Tendenzen erkennbar, ob diese durch das Dashboard eingetreten sind, war in diesem kurzen Zeitraum nicht eindeutig feststellbar. Was aber klar erkennbar war, war die Wahl der Metriken, da durch diese viele Schwachstellen sichtbar gemacht werden konnten.</p> <p>Ergänzend wurde eine <b>qualitative Evaluierung</b> in Form von Interviews durchgeführt. Interviewt wurden dabei eine Entwicklerin, der Product Owner und der Scrum Master. Auffallend dabei war, dass jeder der Drei das Dashboard bereits für seine Zwecke zu nutzen wusste. Während der Product-Owner das Dashboard mehr für Werbezwecke genutzt hat, um andere Abteilungen auf die Möglichkeit von Metriken aufmerksam zu machen, hat es der Scrum-Master eher für die Langzeit-Sicht des Teams genutzt und seinen Fokus auf die Verbesserung des Prozesses gelegt. Die Entwicklerin wiederum hatte den Fokus auf die kurzfristigen Metriken, wie den Bug Count, um schnell auf Probleme reagieren zu können.</p>
Vorgehensweise	Schlussfolgerungen
<p>Es wurde eine Fallstudie, wie in „Agile Metrics In Action“<sup>2</sup> beschrieben, mit einem Scrum Team durchgeführt und dabei in folgenden Schritten vorgegangen:</p> <ol style="list-style-type: none"> <li>1. Metriken ermitteln Die Metriken wurden über die sogenannte GQM (Goal-Question-Metric)-Methode, bei der die Daten der letzten Sprint Retrospektiven ausgewertet wurden, und ergänzend über eine Umfrage im Team ermittelt.</li> <li>2. Software erstellen Die im ersten Schritt ermittelten Metriken wurden über eine eigens erstellte Software ermittelt, aufbereitet und in einem Elastic Stack gespeichert und dargestellt.</li> <li>3. Ergebnisse evaluieren Das Scrum Team nutzte diese Metriken über einen Zeitraum von nicht ganz drei Sprints in den Retrospektiven, um eine bessere Einsicht in den Entwicklungsprozess zu bekommen.</li> </ol>	<p>In dieser Arbeit konnte in einer Fallstudie gezeigt werden, dass mithilfe der GQM-Methodik, ergänzt durch eine Umfrage im entsprechenden Scrum-Team, Metriken ermittelt werden können, die es ermöglichen, die Schwachstellen in einem Produkt und im agilen Prozess in Zahlen zu fassen. Die entwickelte Software hilft dabei, die Daten aus den unterschiedlichen Systemen im Entwicklungsprozess als Metriken aufzubereiten und zu speichern. Für einen einfachen Zugang und eine uneingeschränkte Erweiterbarkeit wurde der Quellcode der Software unter der quellöffentlichen MIT-Lizenz veröffentlicht. Bei der Visualisierung von Metriken bietet Kibana eine geeignete Plattform, um aus den gespeicherten Metriken einfache Dashboards mit unterschiedlichen Visualisierungen bereitzustellen. Durch den Einsatz der entwickelten Software und der vorgestellten Modelle zur Identifizierung von relevanten Metriken, kann die Qualität in einem agilen Team dadurch erhöht werden, dass Qualitätsprobleme durch Metriken sichtbar gemacht und in den Retrospektiven Gegenmaßnahmen dafür getroffen werden können.</p>
Umsetzung	Quellen
<p>Es wurde eine Java-Software erstellt, die eigenständig betrieben werden kann und Daten von den einzelnen Systemen im Entwicklungsprozess abholen, aufbereiten und als Metriken in einer Datenbank speichern kann. In der Fallstudie waren es die Systeme Jira, Bitbucket, SonarQube und zur Speicherung und Visualisierung diente ein Elastic Stack.</p>	<p><sup>1</sup> vgl. Rolf Dräther, Holger Koschek und Carsten Sahling. Scrum: kurz &amp; gut. 1. Auflage. O'Reillys Taschenbibliothek. Beijing Cambridge Farnham Köln Sebastopol, Tokyo: O'Reilly, 2013. isbn: 978-3-86899-833-7, S.14ff  <sup>2</sup> Christopher W. H. Davis. Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams. 1st. Greenwich, CT, USA: Manning Publications Co., 2015. isbn: 978-1-61729-248-4</p>

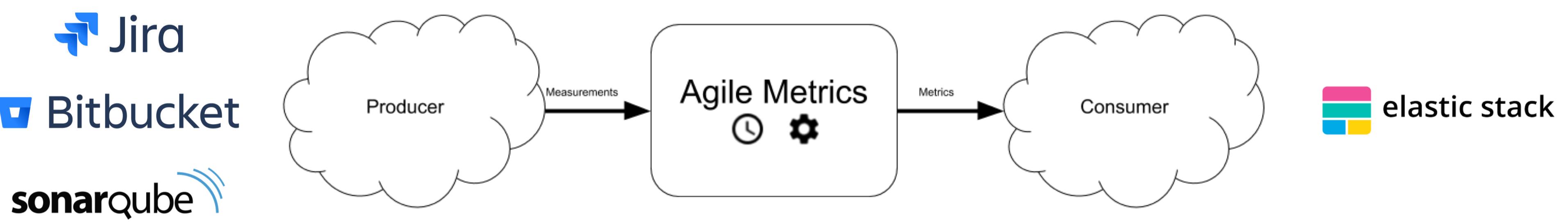


Die entwickelte Software sammelt Metriken in den Systemen entlang des Entwicklungsprozesses und speichert diese aufbereitet zur Visualisierung ab.

# Agile Metriken

## Qualitätssicherung in agilen Teams

<h3>Zielsetzung</h3> <p>„Scrum ist eine Implementierung der Theorie des Empirismus, der zufolge Wissen auf Erfahrung beruht und Entscheidungen auf der Grundlage dieses Wissens getroffen werden. Dazu sind drei Voraussetzungen zu schaffen:</p> <ol style="list-style-type: none"> <li>1. Transparenz</li> <li>2. Überprüfung</li> <li>3. Anpassung“<sup>1</sup></li> </ol> <p>Ziel dieser Arbeit war es, Metriken zu ermitteln, die Qualitätsprobleme im Entwicklungsprozess oder im Softwareprodukt quantitativ abbilden können und somit der Forderung nach Überprüfung gerecht werden.</p>		<h3>Evaluierung</h3> <p>Zur <b>quantitativen Evaluierung</b> wurden die Metriken beobachtet. Es waren nach diesen drei Sprints bereits Tendenzen erkennbar, ob diese durch das Dashboard eingetreten sind, war in diesem kurzen Zeitraum nicht eindeutig feststellbar. Was aber klar erkennbar war, war die Wahl der Metriken, da durch diese viele Schwachstellen sichtbar gemacht werden konnten.</p> <p>Ergänzend wurde eine <b>qualitative Evaluierung</b> in Form von Interviews durchgeführt. Interviewt wurden dabei eine Entwicklerin, der Product Owner und der Scrum Master. Auffallend dabei war, dass jeder der Drei das Dashboard bereits für seine Zwecke zu nutzen wusste. Während der Product-Owner das Dashboard mehr für Werbezwecke genutzt hat, um andere Abteilungen auf die Möglichkeit von Metriken aufmerksam zu machen, hat es der Scrum-Master eher für die Langzeit-Sicht des Teams genutzt und seinen Fokus auf die Verbesserung des Prozesses gelegt. Die Entwicklerin wiederum hatte den Fokus auf die kurzfristigen Metriken, wie den Bug Count, um schnell auf Probleme reagieren zu können.</p>
<h3>Vorgehensweise</h3> <p>Es wurde eine Fallstudie, wie in "Agile Metrics In Action"<sup>2</sup> beschrieben, mit einem Scrum Team durchgeführt und dabei in folgenden Schritten vorgegangen:</p> <ol style="list-style-type: none"> <li>1. Metriken ermitteln Die Metriken wurden über die sogenannte GQM (Goal-Question-Metric)-Methode, bei der die Daten der letzten Sprint Retrospektiven ausgewertet wurden, und ergänzend über eine Umfrage im Team ermittelt.</li> <li>2. Software erstellen Die im ersten Schritt ermittelten Metriken wurden über eine eigens erstellte Software ermittelt, aufbereitet und in einem Elastic Stack gespeichert und dargestellt.</li> <li>3. Ergebnisse evaluieren Das Scrum Team nutzte diese Metriken über einen Zeitraum von nicht ganz drei Sprints in den Retrospektiven, um eine bessere Einsicht in den Entwicklungsprozess zu bekommen.</li> </ol>	<p>langfristige Metriken auf dem Team-Dashboard in Kibana</p>	
<h3>Umsetzung</h3> <p>Es wurde eine Java-Software erstellt, die eigenständig betrieben werden kann und Daten von den einzelnen Systemen im Entwicklungsprozess abholen, aufbereiten und als Metriken in einer Datenbank speichern kann. In der Fallstudie waren es die Systeme Jira, Bitbucket, SonarQube und zur Speicherung und Visualisierung diente ein Elastic Stack.</p>		<h3>Schlussfolgerungen</h3> <p>In dieser Arbeit konnte in einer Fallstudie gezeigt werden, dass mithilfe der GQM-Methodik, ergänzt durch eine Umfrage im entsprechenden Scrum-Team, Metriken ermittelt werden können, die es ermöglichen, die Schwachstellen in einem Produkt und im agilen Prozess in Zahlen zu fassen. Die entwickelte Software hilft dabei, die Daten aus den unterschiedlichen Systemen im Entwicklungsprozess als Metriken aufzubereiten und zu speichern. Für einen einfachen Zugang und eine uneingeschränkte Erweiterbarkeit wurde der Quellcode der Software unter der quellöffentlichen MIT-Lizenz veröffentlicht. Bei der Visualisierung von Metriken bietet Kibana eine geeignete Plattform, um aus den gespeicherten Metriken einfache Dashboards mit unterschiedlichen Visualisierungen bereitzustellen. Durch den Einsatz der entwickelten Software und der vorgestellten Modelle zur Identifizierung von relevanten Metriken, kann die Qualität in einem agilen Team dadurch erhöht werden, dass Qualitätsprobleme durch Metriken sichtbar gemacht und in den Retrospektiven Gegenmaßnahmen dafür getroffen werden können.</p>
<h3>Quellen</h3> <p><sup>1</sup> vgl. Rolf Dräther, Holger Koschek und Carsten Sahling. Scrum: kurz &amp; gut. 1. Auflage. O'Reillys Taschenbibliothek. Beijing Cambridge Farnham Köln Sebastopol, Tokyo: O'Reilly, 2013. isbn: 978-3-86899-833-7, S.14ff <sup>2</sup> Christopher W. H. Davis. Agile Metrics in Action: Measuring and Enhancing the Performance of Agile Teams. 1st. Greenwich, CT, USA: Manning Publications Co., 2015. isbn: 978-1-61729-248-4</p>	<p>The diagram illustrates the data flow. On the left, three cloud icons represent different systems: Jira (blue square with white arrows), Bitbucket (blue square with white branch), and sonarQube (blue square with white wave). Arrows point from these clouds to a central rectangular box labeled 'Agile Metrics' containing a clock and gear icon. An arrow points from 'Agile Metrics' to another cloud icon on the right labeled 'Consumer'. Below the diagram, a caption reads: "Die entwickelte Software sammelt Metriken in den Systemen entlang des Entwicklungsprozesses und speichert diese aufbereitet zur Visualisierung ab."</p>	



Die entwickelte Software sammelt Metriken in den Systemen entlang des Entwicklungsprozesses und speichert diese aufbereitet zur Visualisierung ab.