



catworkx – Atlassian Solutions & Services

# Modellierung von Teamabhängigkeiten in SAFe®

Betrachtung verschiedener Umsetzungsmöglichkeiten in der Atlassian Toolchain

Christoph Piotrowski, März 2020



Platinum  
Solution Partner  
ENTERPRISE

# INHALT

<b>Einführung</b> .....	<b>3</b>
Über den Autor .....	3
<b>Skalierte Agilität</b> .....	<b>4</b>
<b>Das Scaled Agile Framework (SAFe®)</b> .....	<b>5</b>
Das Framework im Überblick .....	5
Programm-Inkremente .....	5
Team-Abstimmungen im „PI Planning“ .....	6
<b>Team-Abhängigkeiten in Jira Align</b> .....	<b>7</b>
Jira Align .....	7
Alle Teams im Blick mit dem Program Board .....	8
Den Status erfassen mit Dependency Reports .....	9
<b>Zusammenarbeit von Teams in Jira Software</b> .....	<b>10</b>
Grundsätzliche Aspekte von Jira Software von Atlassian .....	10
Vorgangsbeziehungen visualisieren mit „Easy Agile Programs“ und „BigPicture“ .....	11
Der Dependency-Filter von „Advanced Roadmaps for Jira“ .....	12
<b>Ideen aus dem Teamworkx Blueprint für einen skalierten, agilen Ansatz nach SAFe®</b> .....	<b>13</b>
Der „Teamworkx Scaling Agile Blueprint“ für SAFe® .....	13
Dependency als Vorgangstyp auf Programmebene .....	13
Erstellen von Dependencies mit dem „Teamworkx Issue Picker for Jira“ .....	14
Automatisierter Sprint-Vergleich .....	15
Das Dependency-Dashboard .....	15
<b>Zusammenfassung</b> .....	<b>16</b>
<b>Literaturverzeichnis</b> .....	<b>16</b>
Rechtliche Anmerkungen .....	16
<b>Über catworkx</b> .....	<b>17</b>

# Einführung

Bei großen agilen Entwicklungsprojekten ist die reibungslose Planung und Abstimmung von teamübergreifenden Aufgaben ein Schlüsselfaktor. Ist ein Team in größeren Entwicklungsprogrammen zunehmend abhängig von Zulieferungen anderer Teams, müssen Methoden definiert werden, die ein reibungsloses Zusammenspiel untereinander sicherstellen.

Scaled Agile Inc. greift dieses Problem im Rahmen des Scaled Agile Framework (SAFe®) auf und bietet auf der konzeptionellen Ebene Vorschläge, wie sich Teams in großen Entwicklungsprojekten abstimmen können. [1]

Bei der toolseitigen Unterstützung agiler Methoden ist Jira Software® von Atlassian™, insbesondere auf Teamebene, weit verbreitet. Aber auch im skalierten, agilen Bereich gibt es darauf aufbauende oder verbundene Lösungen, um die Zusammenarbeit zwischen den Teams effektiv zu gestalten. [2]

Im Folgenden wird zuerst erläutert, wie sich Teams nach SAFe® im sogenannten „PI Planning“ (Program Increment Planning) auf gemeinsame Ziele festlegen, Abhängigkeiten in den Teamplänen feststellen und diese besprechen können.

Im nächsten Schritt werden toolbasierte Ansätze dargestellt, um diese Abhängigkeiten abzubilden. Nach der Darstellung von Jira Align, Jira Software und deren Ergänzungen werden abschließend weitere Vorschläge für die Verbesserung der Teamzusammenarbeit aus dem „Teamworkx Scaling Agile Blueprint“ für SAFe® gemacht.

## Über den Autor



**Christoph Piotrowski** ist bei catworkx verantwortlich für den Bereich der Prozessberatung. Als zertifizierter SAFe® Program Consultant (SPC), Scrum Master, Atlassian Authorized Trainer und Certified Jira Administrator berät er Atlassian-Kunden im Bereich der skalierten agilen Softwareentwicklung. Er verfügt über eine mehr als zehnjährige Erfahrung als Team- und Projektleiter in der Halbleiter- und Automotive-SW-Entwicklung und besitzt ein PMP-Zertifikat von PMI.

# Skalierte Agilität

Der Einsatz agiler Softwareentwicklungsmethoden geht auf Ideen und Vorschläge zurück, die bereits in den 1990er-Jahren formuliert wurden. Viele Ideen und Erkenntnisse wurden unter dem Begriff „Scrum“ von Jeff Sutherland und Ken Schwaber zusammengefasst und stehen heute als „Scrum Guide“ Entwicklungsteams zur Verfügung. [3] Scrum folgt den Werten der agilen Softwareentwicklung, die 2001 im „Agilen Manifest“ in Form von vier Leitsätzen und zwölf Prinzipien formuliert wurden. [4]

Auf dieser Grundlage haben heute vor allem Scrum und Scrum-basierte Vorgehensmodelle weitreichenden Einzug in die Softwareentwicklung gehalten. Hauptgründe für die

Einführung agiler Methoden sind kürzere Zyklen in der Softwareauslieferung und eine bessere Fähigkeit, sich geänderten Rahmenbedingungen und Prioritäten anzupassen. [5]

Während Scrum auf der Teamebene in vielen Unternehmen etabliert ist, treten dennoch oft Fragen auf, wenn es um die teamübergreifende Erstellung größerer Softwarepakete geht: Wie werden die Team-Backlogs übergreifend priorisiert? Wie sollen sich Teams, die möglichst autonom arbeiten, effektiv abstimmen? Wie wird sichergestellt, dass die Ziele der einzelnen Teams einer gemeinsamen Ausrichtung folgen und die entstehenden Softwarepakete zeitlich in der notwendigen Reihenfolge entstehen? Ver-

schiedene Methoden können Unternehmen helfen, ihre agilen Arbeitsweisen zu skalieren. Bekannte Ansätze sind „Scrum of Scrums“, „Large Scale Scrum (LeSS)“, das „Spotify Model“, „Disciplined Agile Delivery (DAD)“ und das „Scaled Agile Framework (SAFe®)“.

SAFe® ist dabei laut „13th Annual State of Agile Report“ mit einem Anteil von 30 Prozent das beliebteste Framework [5].

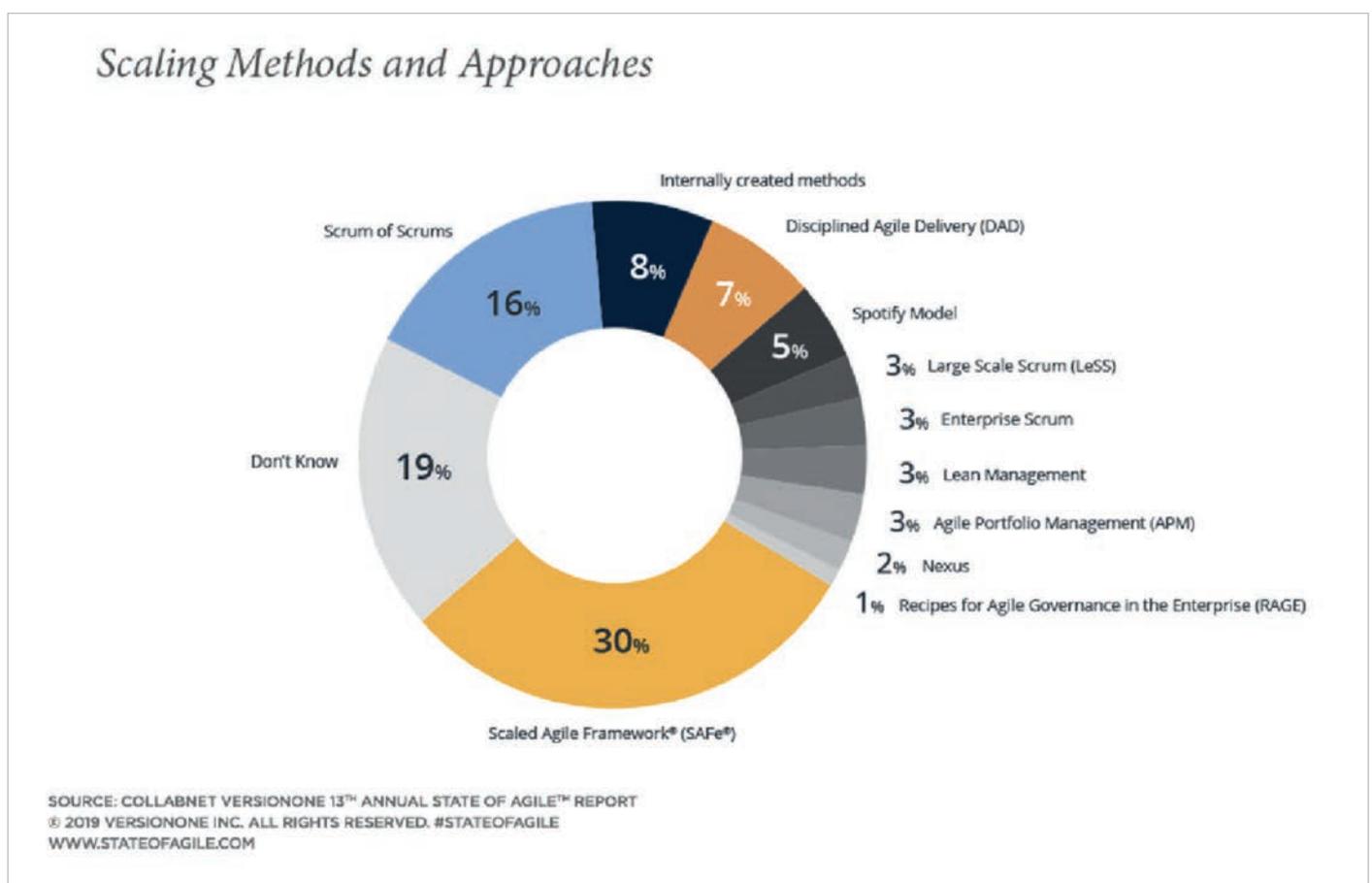


Abb. 1: Scaling Methods and Approaches – 13th State of Agile Report

# Das Scaled Agile Framework (SAFe®)

## Das Framework im Überblick

SAFe® wurde für die Skalierung von Agilität in Unternehmen entwickelt. Das Framework ermöglicht größeren Unternehmen und Organisationen die strategische Planung und Steuerung mit dem Ziel, Innovationen voranzutreiben, Kundenanforderungen schnell und mit einer guten Produktqualität zu erfüllen sowie Ineffizienz zu reduzieren. Durch seinen ganzheitlich ausgerichteten Ansatz ermöglicht es agile Arbeitsweisen über alle Teams hinweg und verhindert die Bildung von Silos in Unternehmen. Ausgehend von der strategischen Ausrichtung über das Management bis hin zur Produktion und Auslieferung möchte SAFe® den

Projekt- und Prozessrahmen bieten, mit dem sich größere Unternehmen den Herausforderungen der Digitalisierung stellen können. Auch die unternehmensweite Komplexität der unterschiedlichen Prozesse können mit SAFe® abgebildet und gesteuert werden. Die Basis von SAFe® sind, in Anlehnung an das „Agile Manifest“, Werte und Prinzipien, die verstärkt Skalierbarkeit, Systemisches Denken und Business-Agilität in den Blickpunkt nehmen. [6]

## Programm-Inkremente

Das Programm-Inkrement (PI) ist ein entscheidendes Element bei SAFe®, mit dessen Hilfe das siebte Prinzip („Apply cadence, synchronise with cross domain planning“) in die Praxis umgesetzt werden soll. SAFe® beschreibt das PI als einen Zeitabschnitt, in welchem ein Agile Release Train (ART) inkrementelle Werte, in Form von lauffähiger Software oder Systeme, liefert. Ein ART besteht dabei aus einer Anzahl von ungefähr 50–125 Personen. Das Programm-Inkrement umfasst dabei zeitlich übergreifend mehrere (~5) Team-Iterationen. Somit kann das PI auf der Ebene des ART als Pendant zu der Iteration (oder Sprint) auf der Teamebene gesehen werden. [7]

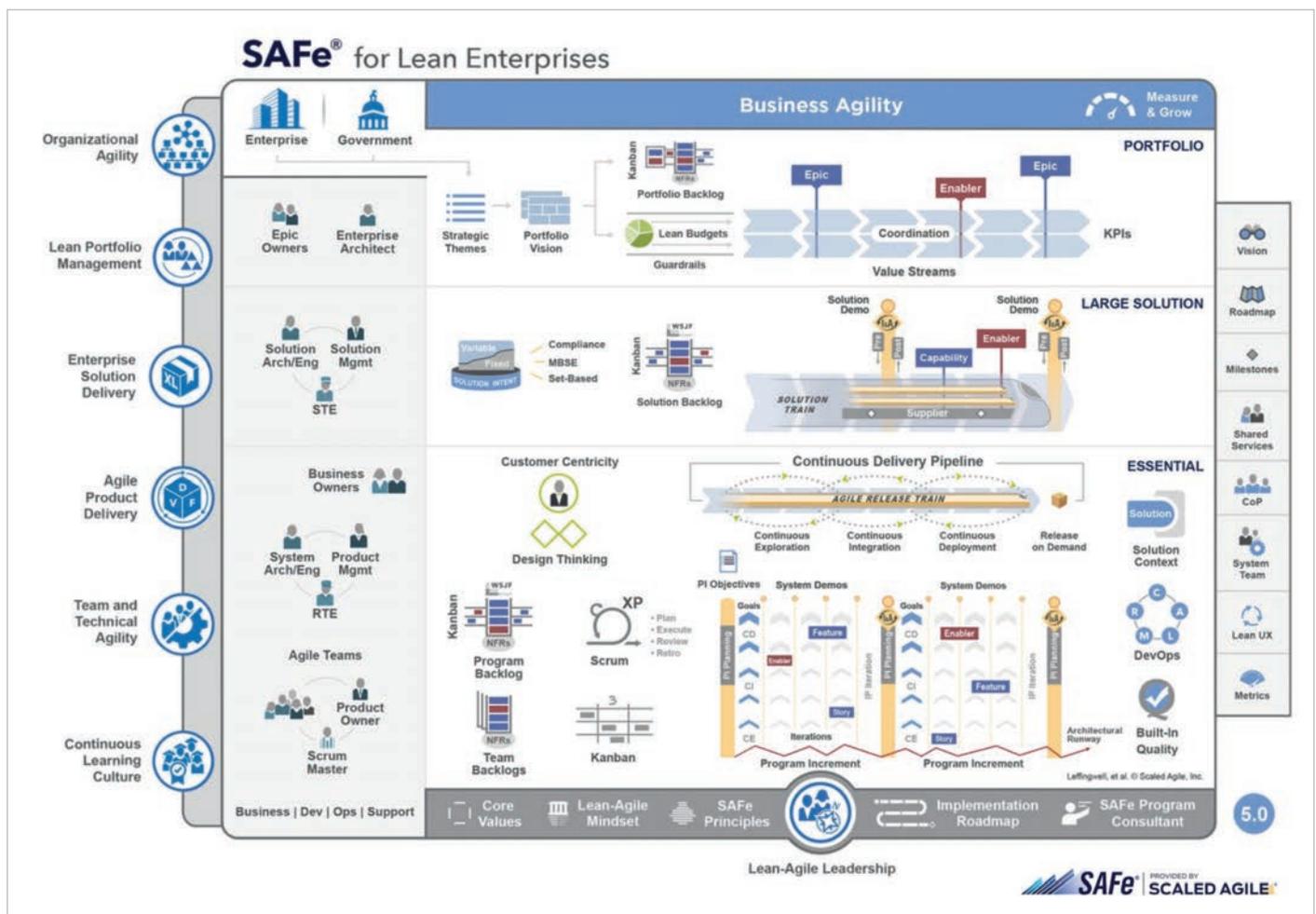


Abb. 2: SAFe® im Überblick, das „Big Picture“

## Das Scaled Agile Framework (SAFe®)

### Team-Abstimmungen im „PI Planning“

SAFe® gibt eine Reihe von Empfehlungen für die Planung von Programm-Inkrementen. So wird Wert auf den direkten, persönlichen Austausch gelegt, das Einholen von Team-Zusagen (Commitment) zu der nächsten Iteration und das Identifizieren von Abhängigkeiten zwischen den Arbeitspaketen der einzelnen Teams.

Ein elementarer Baustein bei SAFe® ist das sogenannte „PI Planning“ (Programm Increment Planning – zentrales Synchronisationsmeeting), das kurz vor jedem Programm-Inkrement mit allen Beteiligten des ART durchgeführt wird. Ein solches „PI Planning“ folgt einer bestimmten Agenda und dauert normalerweise zwei Tage. Zentrale Bestandteile sind die „Team Breakouts“, in denen die Teams eigene Pläne erstellen und mit allen Stakeholdern abstimmen können.

Während der „Team Breakouts“ entstehen auf Programmebene die Program Boards: Die einzelnen Teams visualisieren mit Karten die geplanten Lieferzeiten ihrer Features. Abhängigkeiten unter den Features und Teams werden dabei ebenfalls kenntlich gemacht.

Während ein physikalisches Board bei der Planung noch unterstützen kann, benötigen die Teams während der dann folgenden Iterationen eine gemeinsame Informationsquelle, um den Status ihrer Arbeiten auch für die anderen Teams immer aktuell verfügbar zu machen. Zudem sollen neue und sich ändernde Abhängigkeiten zwischen anderen Teams und potenzielle Probleme durch verzögerte Arbeitspakete schnell erkannt werden, damit Probleme in der Featureplanung möglichst frühzeitig erkannt und behoben werden können. [8]

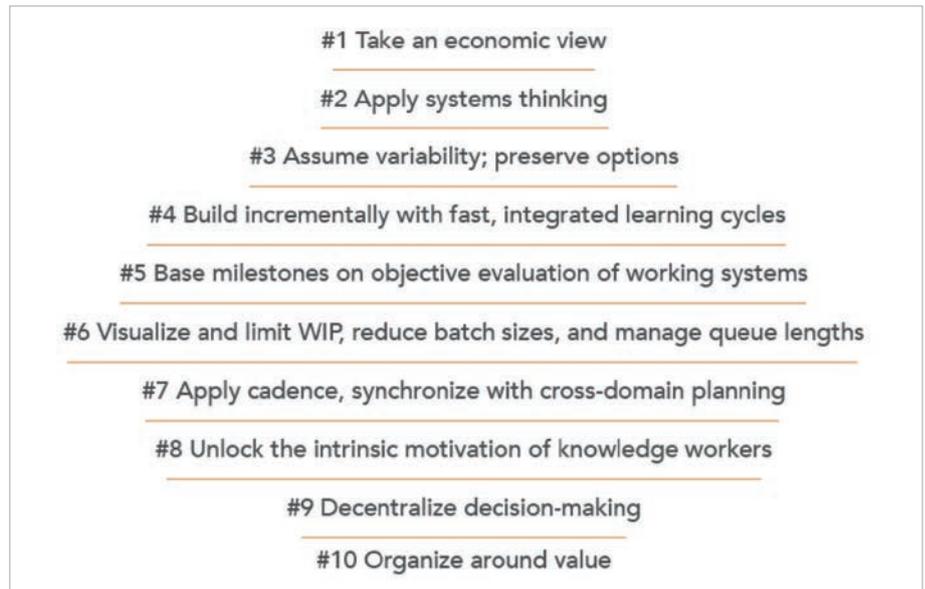


Abb 3: SAFe® Lean - Agile Prozesse



Abb 4: Face-to-Face „PI Planning“



Abb 5: Agenda „PI Planning“

# Team-Abhängigkeiten in Jira Align

## Jira Align

Atlassian bietet mit Jira Align® [9] einen ganzheitlichen Ansatz für die agile Entwicklung von großen Softwareapplikationen in Unternehmen, der die Geschäftsstrategie mit der technischen Umsetzung verbindet. Jira Align erweitert dabei als separates Tool die

Arbeiten der einzelnen Teams in Jira Software, um die Möglichkeiten einer professionellen Arbeitsorganisation in der Programm- und Portfolioebene. Während die Entwickler weiterhin in Jira Software arbeiten, können sie ein weitergehendes Verständnis über die Ziele und Hintergründe ihrer Arbeit erlan-

gen. RTEs, Programm- und Produktmanager dagegen haben mit Jira Align die Möglichkeit, teamübergreifend zu planen und den aktuellen Status ihrer Arbeit im Blick zu behalten.

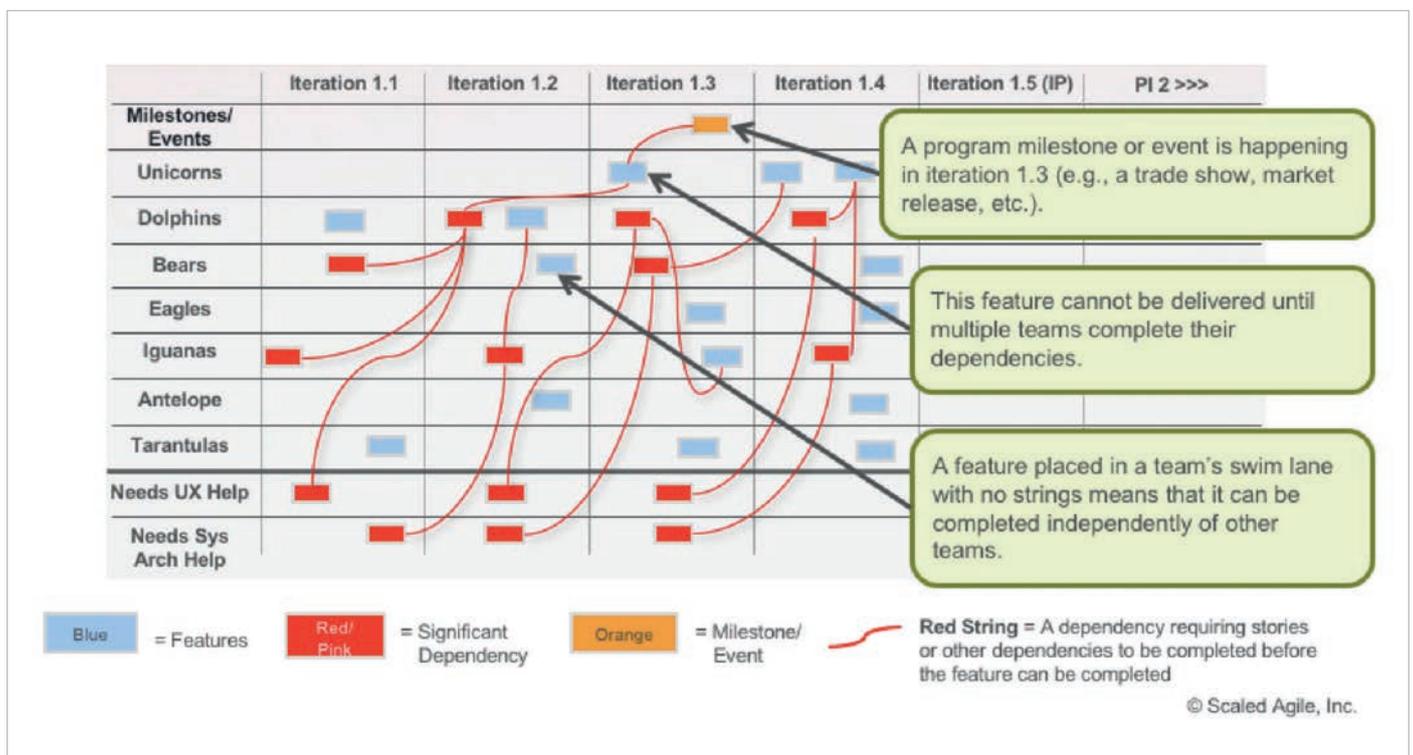


Abb 6: Program Board für die Visualisierung von Features und Abhängigkeiten

## Team-Abhängigkeiten in Jira Align

### Alle Teams im Blick mit dem Program Board

Um in SAFe® „PI Plannings“ zu unterstützen und die dann startenden Iterationen zu unterstützen, bietet Jira Align ein Program Board, welches einen Überblick über die geplanten Features aller Teams des ausgewählten Release Trains gibt. Hervorzuheben ist, dass eine Abhängigkeit („Dependency“) zwischen den Features als eigenes Objekt modelliert ist. Im gezeigten Screenshot sind sechs Iterationen (S23 bis S28) dargestellt.

In den Team-Swimlanes befinden sich übersichtlich angeordnet



Abb. 7: Program Board (Jira Align)

- die Features, die das Team in den Sprints geplant hat
- sowie die Abhängigkeiten zu anderen Teams.

In der Abbildung rechts, sind die Team-Abhängigkeiten rot eingefärbt und zeigen damit einen Planungskonflikt auf. Fährt der Anwender über das Kreuzsymbol, werden die voneinander abhängigen Objekte hervorgehoben.

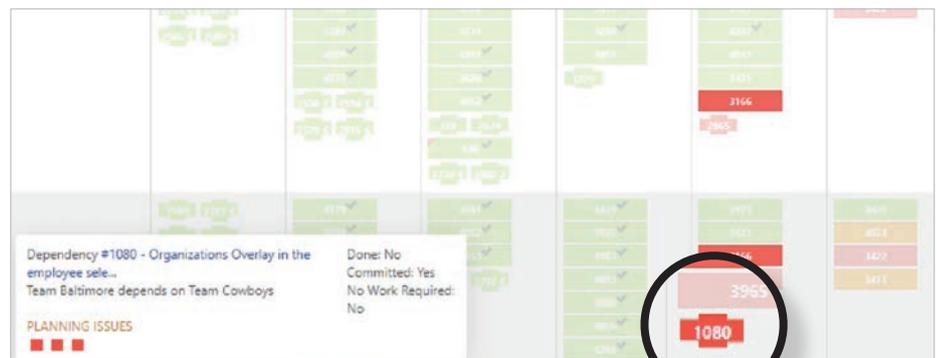


Abb. 8: Dependency Highlighting (Jira Align)

Durch einen Klick auf das Objekt gibt das Tool Auskunft über den Hintergrund des Konfliktes. So könnte es z. B. sein, dass ein Team die Abhängigkeit nicht bestätigt hat, oder das Arbeitspaket zu spät eingeplant wurde.

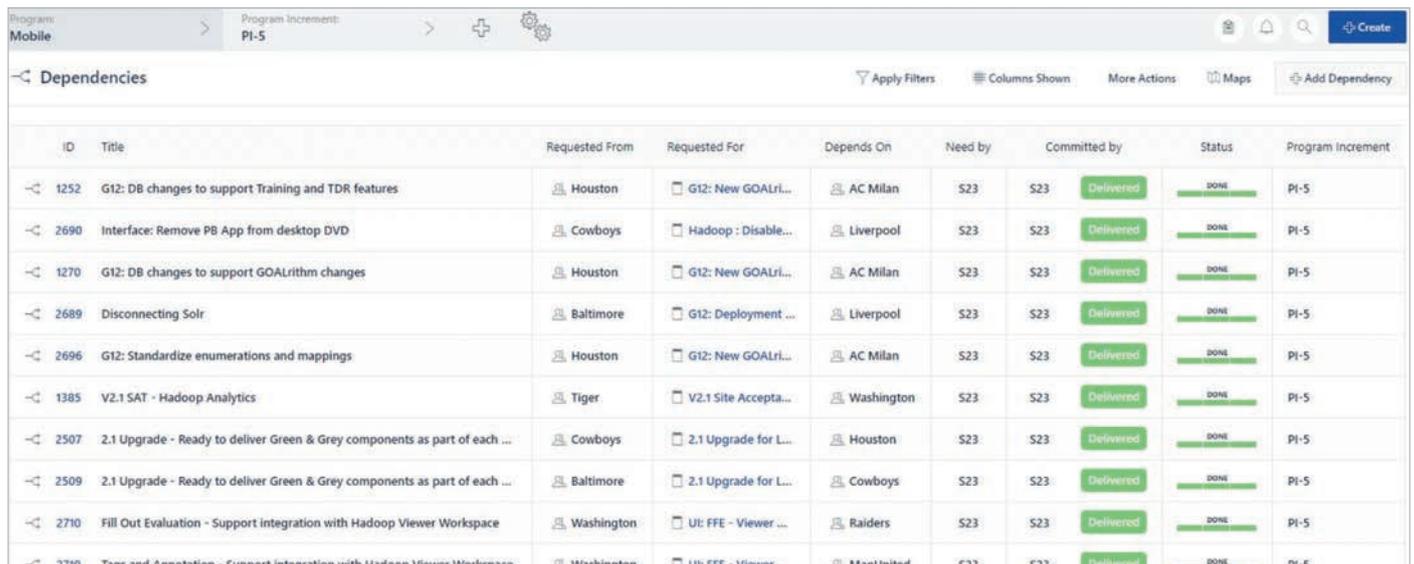
## Team-Abhängigkeiten in Jira Align

### Den Status erfassen mit Dependency Reports

Neben dem Program Board können unterschiedliche „Dependency Reports“ weitere Informationen geben. Listenansichten ermöglichen es beispielsweise einem Release Train Engineer den Status aller Team-Verbindungen im Blick zu behalten.

Eine „Dependency-Matrix“ visualisiert die Anzahl der Verbindungen zwischen Teams in Tabellenform. Mit dem „Dependency Wheel“ lassen sich schnell Teams identifizieren, die in der Zusammenarbeit mit besonders vielen Abhängigkeiten einen besonderen Fokus benötigen.

Alle Reports lassen sich flexibel auf einzelne oder mehrere Programm-Inkmente, Teams und Portfolios eingrenzen und stellen damit ein flexibles Hilfsmittel für unterschiedliche Rollen im Unternehmen dar. Diese können helfen, Engpässe bei der Zusammenarbeit von Teams zu identifizieren.



ID	Title	Requested From	Requested For	Depends On	Need by	Committed by	Status	Program Increment
1252	G12: DB changes to support Training and TDR features	Houston	G12: New GOALr...	AC Milan	S23	S23	Delivered	PI-5
2690	Interface: Remove PB App from desktop DVD	Cowboys	Hadoop : Disab...	Liverpool	S23	S23	Delivered	PI-5
1270	G12: DB changes to support GOALrithm changes	Houston	G12: New GOALr...	AC Milan	S23	S23	Delivered	PI-5
2689	Disconnecting Solr	Baltimore	G12: Deployment ...	Liverpool	S23	S23	Delivered	PI-5
2696	G12: Standardize enumerations and mappings	Houston	G12: New GOALr...	AC Milan	S23	S23	Delivered	PI-5
1385	V2.1 SAT - Hadoop Analytics	Tiger	V2.1 Site Accepta...	Washington	S23	S23	Delivered	PI-5
2507	2.1 Upgrade - Ready to deliver Green & Grey components as part of each ...	Cowboys	2.1 Upgrade for L...	Houston	S23	S23	Delivered	PI-5
2509	2.1 Upgrade - Ready to deliver Green & Grey components as part of each ...	Baltimore	2.1 Upgrade for L...	Cowboys	S23	S23	Delivered	PI-5
2710	Fill Out Evaluation - Support integration with Hadoop Viewer Workspace	Washington	UI: FFE - Viewer ...	Raiders	S23	S23	Delivered	PI-5
2719	Tags and Annotation - Support integration with Hadoop Viewer Workspace	Washington	UI: FFE - Viewer ...	ManUnited	S23	S23	Delivered	PI-5

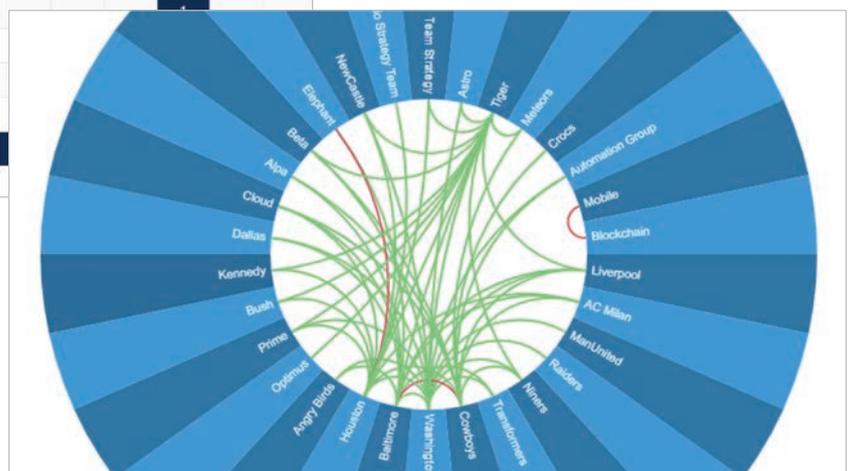
Abb. 9: Dependency-Listen (Jira Align)



	AC Milan	Angry Birds	Baltimore	Cowboys	Dallas	Houston	Liverpool	ManUnited	Niners	Optimus	Raiders	Tiger	Transformers	Washington
AC Milan			2	1										
Alpa			1	1										
Angry Birds														1
Automation Group				1										1
Baltimore		1		2			1							
Beta				1										
Bush						1								
Cloud			1	1		1								
Cowboys	1		3			1	1							
Dallas			1											

Abb. 11: Dependency Wheel (Jira Align)

Abb. 10: Dependency-Matrix (Jira Align)



# Zusammenarbeit von Teams in Jira Software

## Grundsätzliche Aspekte von Jira Software von Atlassian

Jira Software ist nach dem „13th Annual State of Agile Report“ [5] das am häufigsten eingesetzte Tool für die agile Softwareentwicklung.

Jira Software bietet einzelnen, agilen Teams wertvolle Hilfsmittel wie Scrum- und Kanban-Boards, Möglichkeiten des Arbeitens in Sprints und das Verwalten von Produkt- und Team-Backlogs. Jira Software ist nicht zuletzt aufgrund seiner Konfigurier- und Erweiterbarkeit bei vielen Entwicklungsteams beliebt, da es je nach Entwicklungsprojekt flexibel auf die Bedürfnisse des Teams angepasst werden kann.

In der Zusammenarbeit mit anderen Teams bietet Jira Software alle Möglichkeiten, die seine Konfigurierbarkeit mit sich bringt. Aber diese Konfigurierbarkeit erfordert auch

die Einigung auf ein Mindestmaß an einheitlichen Arbeitsabläufen und Regeln, wenn beispielsweise synchrone Sprints, vergleichbare Vorgangstypen und Arbeitsabläufe verwendet werden sollen.

Für das agile Skalieren mit Jira Software sind exemplarisch die folgenden Aspekte genannt, die Jira nicht im Standard vorgibt, bei der Konfiguration aber berücksichtigt werden sollten:

- Teams sollten in einheitlichen Sprints arbeiten, die Sprint-Kadenz muss abgestimmt und entsprechend in den einzelnen Teambereichen konfiguriert werden.
- Es gibt kein standardisiertes „Team“-Feld. Eine Teamdefinition kann über Custom-Felder (z. B. Select-Listen), Projekte, Boards oder Erweiterungen von Drittanbietern erzielt werden.

- Abhängigkeiten zwischen Vorgängen können über sogenannte Verlinkungen erzielt werden. Diese Verlinkungen sind Beziehungsinformationen („blockedby“, „refers to“, „depends on“), die zwar eine Richtung, aber keinen Status haben können (im Gegensatz zu Jira Align, s. o.)

Um die Konfiguration im skalierten, agilen Umfeld zu erleichtern, bietet der „Teamworkx Scaling Agile Blueprint“ für SAFe® eine Kombination aus empfohlenen Einstellungen und ausgewählten Erweiterungen, auf die später eingegangen werden soll. In den folgenden Abschnitten wird insbesondere aufgezeigt, wie Abhängigkeiten unter Vorgängen und Teams modelliert werden können.

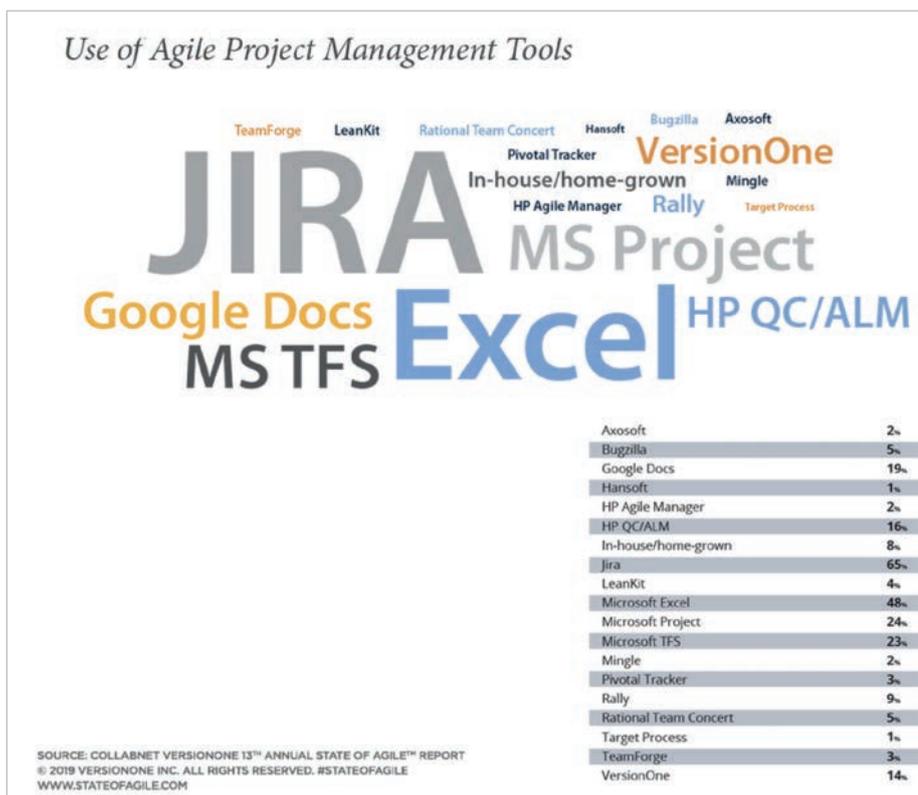


Abb. 12: Verbreitung von agilen Projektmanagement-Tools – 13th Annual State of Agile Report, CollabNet VersionOne

## Zusammenarbeit von Teams in Jira Software

### Vorgangsbeziehungen visualisieren mit „Easy Agile Programs“ und „BigPicture“

Um Program Boards, die während des „PI Plannings“ entstehen, zu digitalisieren oder von Beginn an digital zu erstellen, bieten die Hersteller zweier Apps für Jira eine gute Möglichkeit, Vorgangsbeziehungen anschaulich darzustellen.

„BigPicture“ [10] von der Firma SoftwarePlant ist ein Tool für die Erstellung von Roadmaps

und die Verwaltung von Ressourcen und Risiken im Projekt. Außerdem bietet es eine SAFe®-konforme Möglichkeit, Arbeiten in Programm-Iterationen zu planen und mit „Objectives“ zu versehen. Anhängigkeiten von Arbeitspaketen lassen sich auch teamübergreifend durch farbige Verbindungslinien darstellen. Die Farbe signalisiert dabei, ob die Verbindung der Vorgänge der Planung entspricht (grün), oder ob es ein Planungsrisiko (gelb) oder gar einen Konflikt (rot) gibt. Eine sehr ähnliche Darstellung bietet auch

„Easy Agile Programs for Jira“ [11] von Easy Agile. Die Erweiterung für Jira Software unterstützt ebenfalls das Planen von Sprints und Programm-Inkrementen.

Teams werden dabei aus den Jira-Boards abgeleitet und Beziehungen zwischen den Vorgängen werden wieder entsprechend farblich dem Planungsstand visualisiert.

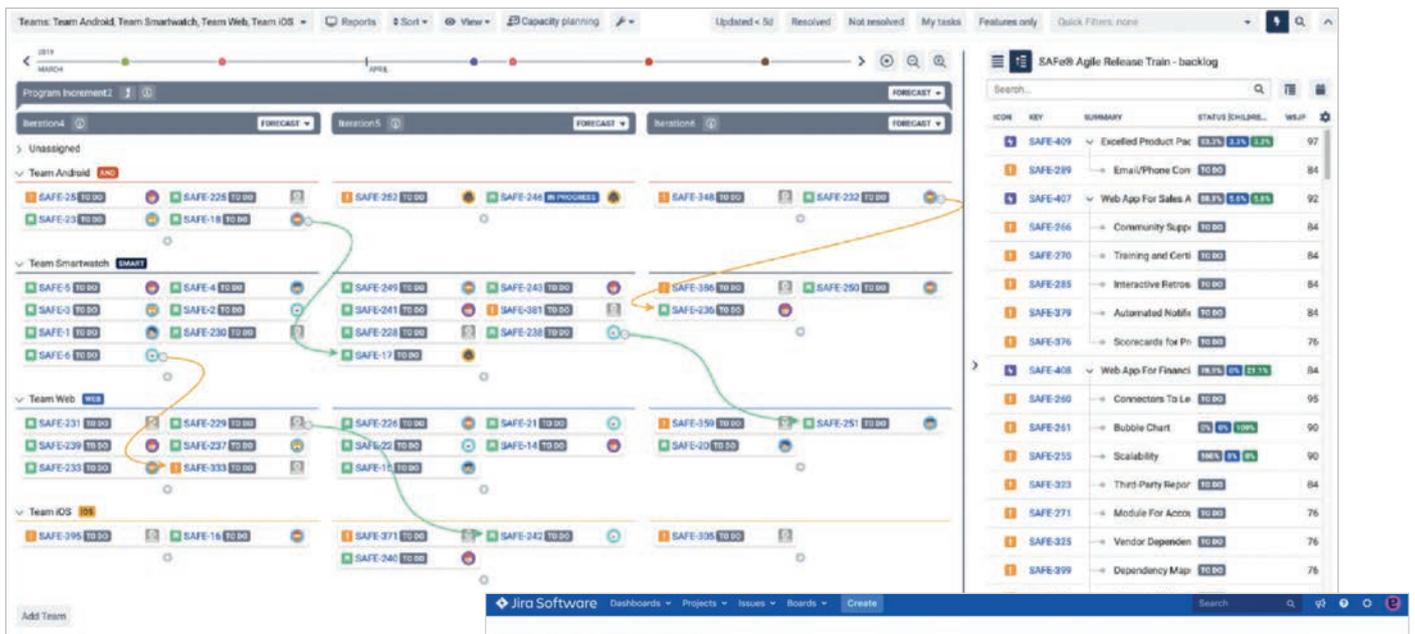


Abb. 13: Boardansicht von „BigPicture“ (SoftwarePlant)

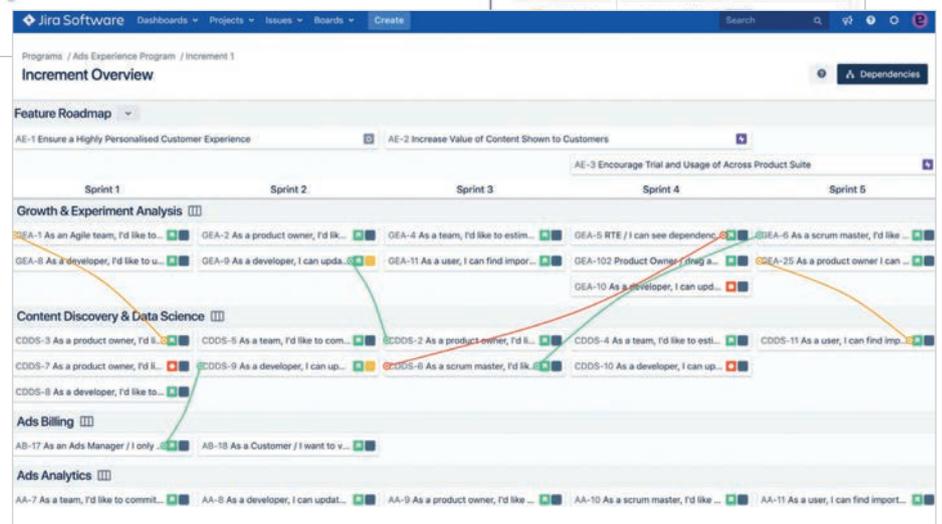


Abb. 14: Übersicht Programm-Inkrement (PI) in „Easy Agile Programs“ (Easy Agile)

## Zusammenarbeit von Teams in Jira Software

### Der Dependency-Filter von „Advanced Roadmaps for Jira“

Eine weitere App „Advanced Roadmaps for Jira“ [12] kommt direkt von Atlassian und ermöglicht eine projekt- und teamübergreifende Darstellung von Vorgängen in Form einer Vor-

gangsliste, kombiniert mit einem Gantt-Chart-Bereich. Vorgangsbeziehungen werden in Form von Markern angedeutet, Details erfährt der Anwender durch einen Klick auf den Marker. Eine gezielte Übersicht aller Abhängigkeiten, die ein Planungsproblem beinhalten, lässt

sich erreichen, in dem der Vorgangsfiler auf „Issues with dependencies“ gestellt und der „Warnings“-Schalter aktiviert wird.

Abbildung 17 zeigt eine solche Trefferliste, gruppiert nach den jeweiligen Teams.

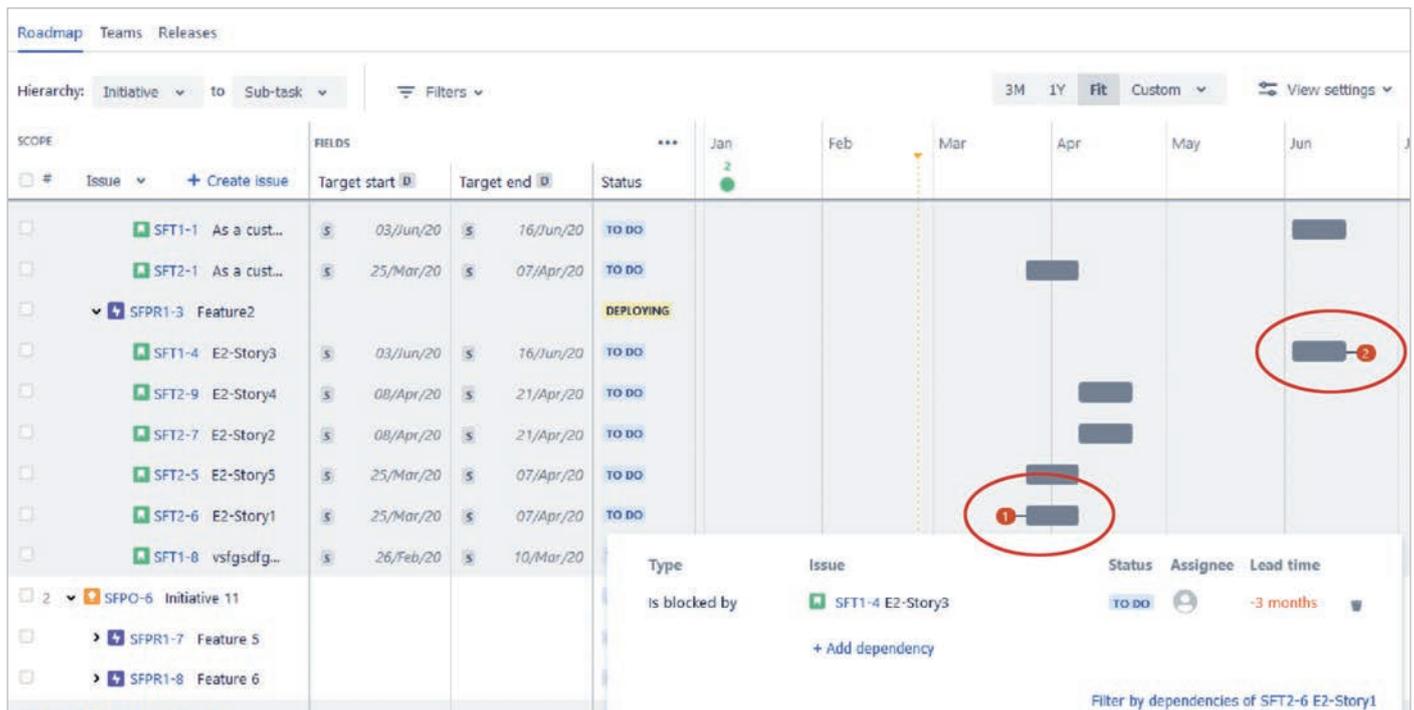


Abb. 15: Dependency-Marker in der Roadmap Ansicht von „Advanced Roadmaps for Jira“ (Atlassian)

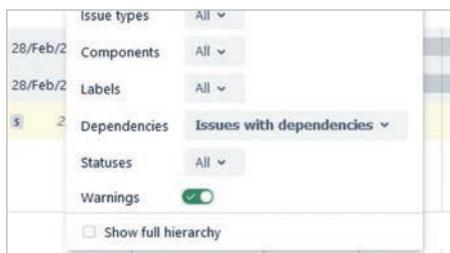


Abb. 16: Filtern nach Dependencies in „Advanced Roadmaps for Jira“

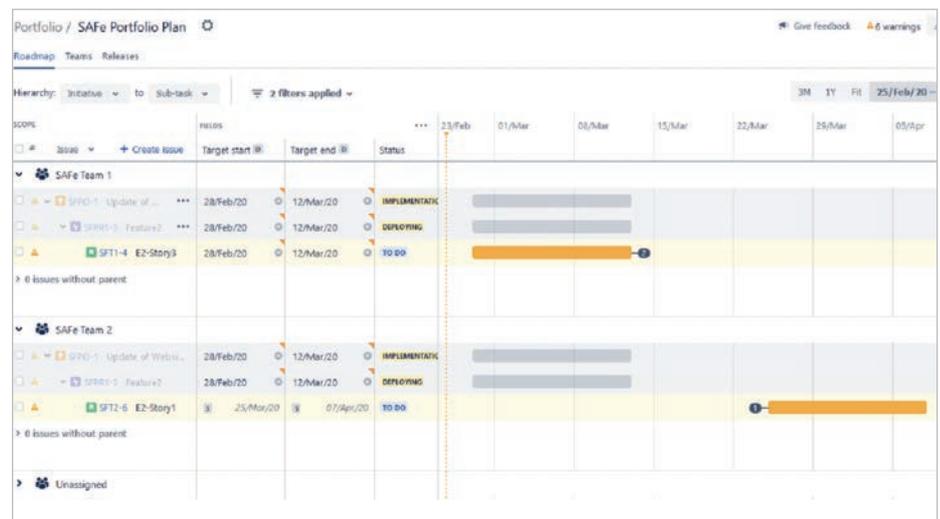


Abb. 17: Liste aller Dependencies mit Warnungen („Advanced Roadmaps for Jira“)

# Ideen aus dem Teamworkx Blueprint für einen skalierten, agilen Ansatz nach SAFe®

## Der „Teamworkx Scaling Agile Blueprint“ für SAFe®

Um die Konfiguration von Jira Software im skalierten, agilen Umfeld zu erleichtern, bietet der „Teamworkx Scaling Agile Blueprint“ für SAFe® eine Kombination aus empfohlenen Einstellungen und ausgewählten Erweiterungen (Apps). Der Blueprint umfasst dabei Vorschläge für das Arbeiten in Jira von der Team- bis zur Portfolioebene sowie Ideen zur Verknüpfung zur Unternehmensstrategie. Zentrale Bestandteile des Blueprints sind neben Jira Software und Confluence die bei-

Weitere alternative Apps, die je nach Kundensituation gut zu einer SAFe®-Implementierung passen können, sind vor allem „BigPicture“ [10] von SoftwarePlant und „Advanced Roadmaps for Jira“ [12] von Atlassian.

## Dependency als Vorgangstyp auf Programmebene

Als grundlegende Konfiguration auf Teamebene wird im Blueprint das Anlegen von separaten Jira-Projekten für die einzelnen Teams empfohlen. Auf dieser Ebene werden die Team-Backlogs verwaltet und Sprints ge-

Stories. Auch wenn häufig ein Epic durch ein einzelnes Team bearbeitet wird, kann es auch Stories für mehrere Teams enthalten. Im „Teamworkx Scaling Agile Blueprint“ für SAFe® befinden sich alle Epics in einem separaten Projekt, auf das alle Teams zugreifen können. Müssen Vorgänge in einer bestimmten Reihenfolge bearbeitet werden, so können diese in Jira beispielsweise mit einem „Blocked by“-Link verknüpft werden. Diese Beziehung ist nicht nur in den betroffenen Stories selbst sichtbar, sondern lässt sich auch mit Apps wie „BigPicture“ oder „Easy Agile Programs for

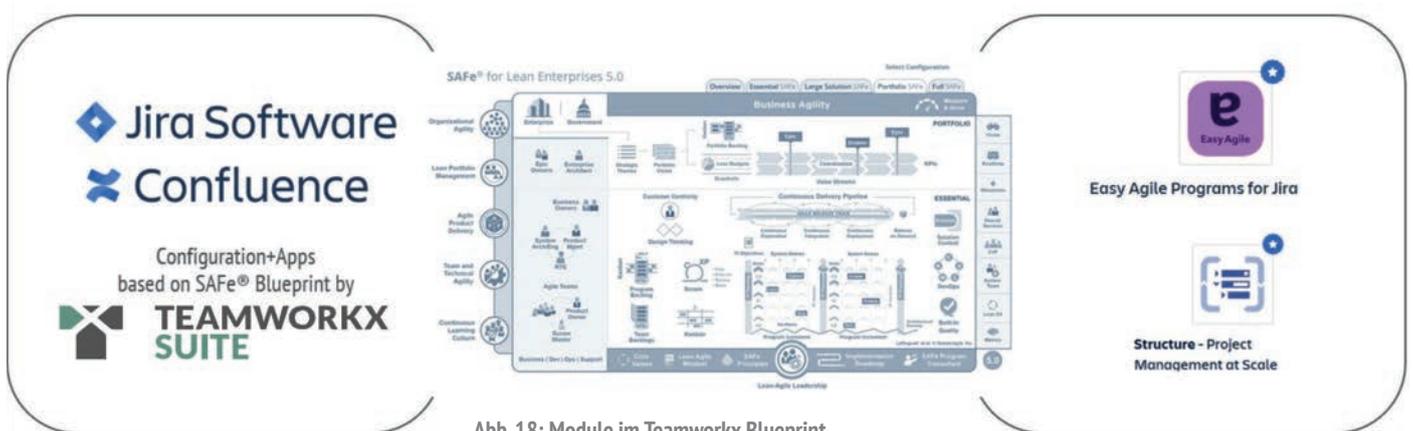


Abb. 18: Module im Teamworkx Blueprint

den Apps „Easy Agile Programs for Jira“ [11] von Easy Agile und „Structure“ [13] von ALM Works. Während mit „Easy Agile Programs for Jira“ die Programm-Inkrement-Planung unterstützt wird, dient „Structure“ einer übersichtlichen Darstellung von Vorgangsmengen für verschiedene Rollen (RTE, Teammitglieder, Produktmanager). Aufgrund seiner umfassenden Konfigurierbarkeit ist es ein flexibles Werkzeug, um kundenspezifische Anpassungen vornehmen zu können.

plant. Den Teams stehen für ihre Arbeiten agile Reports wie „Sprint“ und „Epic Burndown Charts“ zur Verfügung. Auf dieser Ebene werden hauptsächlich die Stories aus dem Team-Backlog durch das Team implementiert.

Auf der SAFe®-Programmebene werden Features im Programm-Backlog priorisiert. Diese Features entsprechen in Jira dem Vorgangstypen „Epic“ und beinhalten die von dem Product Owner und den Teams abgeleiteten

Jira“ auswerten. Für teamübergreifende Verbindungen zwischen Stories verschiedener Teams bietet der „Teamworkx Scaling Agile Blueprint“ für SAFe® die Möglichkeit, „Dependency-Objekte“ auf der Programmebene anzulegen. Hierbei werden die vorhandenen „Blocked by“-Links beibehalten und das zusätzliche Objekt mit einem eigenen Link-Typ („needs“) verbunden. Das Verwenden eines eigenen Vorgangstypen, für das Abbilden von Abhängigkeiten zwischen Teams, bietet

## Ideen aus dem Teamworkx Blueprint für einen skalierten, agilen Ansatz nach SAFe®

einige Vorteile. Der Vorgangstyp „Dependency“ hat einen eigenen Workflow in Jira, der die Abstimmung zwischen den Teams zu dieser Dependency unterstützt und transparent darstellt.

Der dargestellte Workflow macht sichtbar, ob eine Abhängigkeit zwischen Arbeitspaketen der Teams bestätigt wurde oder sich noch in der Klärung befindet. Auf der Programmebene lassen sich mit einfachen Mitteln offene Klärungen auflisten. Eine solche Liste könnte beispielsweise für den RTE im „ART Sync Meeting“ ein nützliches Mittel sein, um offene Abstimmungen anzusprechen und eventuelle Risiken zu identifizieren.

### Erstellen von Dependencies mit dem „Teamworkx Issue Picker for Jira“

Das Anlegen der „Dependency-Objekte“ kann mithilfe der App „Teamworkx Issue Picker for Jira“ [14] bequem über Dropdown-Felder erfolgen. Die relevanten Stories, die in Abhängigkeit zueinanderstehen, werden im Create-Screen von Jira ausgewählt. Beim Erstellen des Vorgangs werden die benötigten „Needs“-Links zwischen den Stories und dem „Dependency-Objekt“ automatisch angelegt.

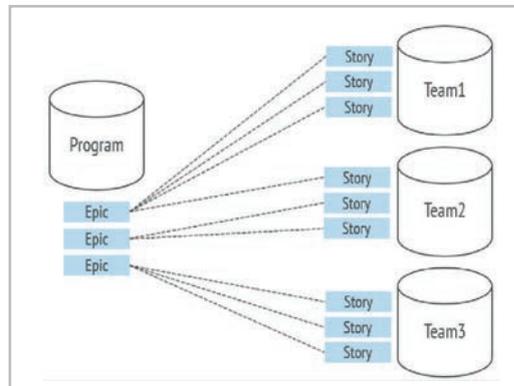


Abb. 19: Schematische Darstellung der Jira-Projekte im Teamworkx Blueprint

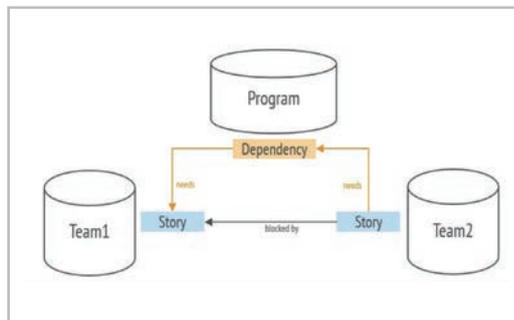


Abb. 20: Link-Schema eines Dependency-Vorgangs im Teamworkx Blueprint

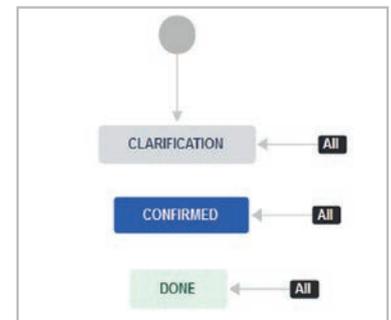


Abb. 21: Workflow eines Dependency-Vorgangs im Teamworkx Blueprint

Abb. 22: Erstellen eines Dependency-Objektes im Teamworkx Blueprint

## Ideen aus dem Teamworkx Blueprint für einen skalierten, agilen Ansatz nach SAFe®

### Automatisierter Sprint-Vergleich

Spezielle Custom-Felder aus der Jira-Blueprint-Konfiguration werten die Sprint-Informationen der verlinkten Stories aus und ermitteln eventuelle Terminkonflikte. Das Ergebnis wird in einem Feld in der Detailansicht des Vorgangs dargestellt.

### Das Dependency-Dashboard

Ein exemplarisches Dashboard aus dem „Teamworkx Scaling Agile Blueprint“ für SAFe® zeigt, wie anhand der beschriebenen Konfigurationen die Abhängigkeiten unter den Teams über Dashboard Gadgets übersichtlich dargestellt werden können.

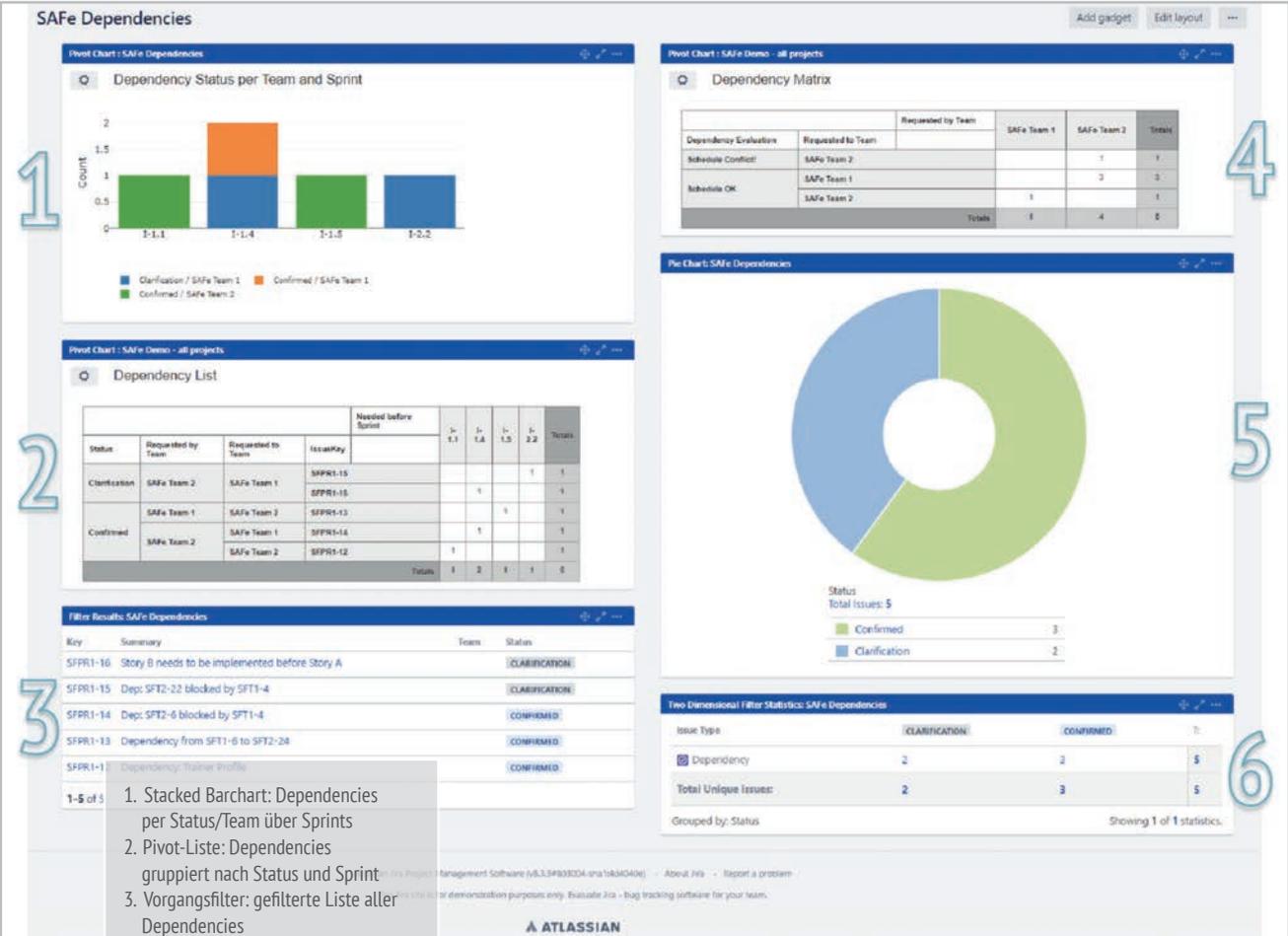


SAFe Program 1 / SFPR1-12  
**Dependency: Trainer Profile**

**Details**

Type:  Dependency  
 Priority:  Medium  
 Status: **CONFIRMED** (View Workflow)  
 Resolution: Unresolved  
 Affects Version/s: None  
 Fix Version/s: None  
 Labels: None  
 Needed by Story: SFT2-10: As a customer I want to be able to book trainings with a mobile device  
 Needs Story: SFT2-1: As a customer I want to see the trainer profile linked to a class  
 Requested by Team: SAFe Team 2  
 Requested to Team: SAFe Team 2  
 Needed before Sprint: I-1.1  
 Planned for Sprint: I-1.4  
 Dependency Evaluation: **Schedule Conflict!**

Abb. 23: Automatisierter Sprintvergleich von verlinkten Stories („Teamworkx Scaling Agile Blueprint“ für SAFe®)



**SAFe Dependencies**

1. **Stacked Barchart: Dependencies per Status/Team über Sprints**

2. **Pivot-Liste: Dependencies gruppiert nach Status und Sprint**

3. **Vorgangsfiler: gefilterte Liste aller Dependencies**

4. **Dependency-Matrix: Team-Abhängigkeiten zusammengefasst in einer Team-Matrix**

5. **Dependency Status Pie Chart**

6. **Zweidimensionaler Filter mit Status**

Abb. 24: Beispiel eines Dependency-Dashboards (Teamworkx Blueprint)

# Zusammenfassung

Die Atlassian Toolchain bietet vielfältige Möglichkeiten der Unterstützung für Entwicklungsprojekte im skalierten, agilen Unternehmen. Wird gezielt die teamübergreifende Abstimmung betrachtet, bietet Jira Align umfangreiche Möglichkeiten, Abhängigkeiten unter den Teams zu definieren und mithilfe verschiedener Reports im Blick zu behalten. Jira Software bietet mithilfe zusätzlicher Apps wie „BigPicture“ von SoftwarePlant, „Easy Agile Programs for Jira“ von Easy Agile und „Advanced Roadmaps for Jira“ von

Atlassian weitere Möglichkeiten, Abhängigkeiten unter Teams zu visualisieren und so z. B. SAFe®-Events wie das „PI Planning“ zu unterstützen. Weitere Möglichkeiten des „Dependency-Managements“ in großen Softwareentwicklungsprojekten wurden anhand von Beispielen aus dem „Teamworkx Scaling Agile Blueprint“ für SAFe® aufgezeigt.

Aufgrund der Komplexität einer SAFe®-Implementierung kann anhand der hier dargestellten Details keine abschließende

Tool-Auswahl empfohlen werden. Aufgezeigt wurden dagegen verschiedene Möglichkeiten, die bei einer Entscheidungsfindung für Produkte oder Konfiguration vor dem Hintergrund der bereits vorhandenen Tool-Landschaft, der konkreten Zielsetzung und der Entwicklungsteamgröße, unterstützen können.

# Literaturverzeichnis

- [1] Scaled Agile, Inc., „Scaled Agile Framework“, [Online]. Available: <https://www.scaledagileframework.com>
- [2] Atlassian, „Atlassian Website“, [Online]. Available: [www.atlassian.com](http://www.atlassian.com)
- [3] K. Schwaber und J. Sutherland, „The Scrum Guide(TM)“, 2017. [Online]. Available: <https://www.scrumguides.org/scrum-guide.html>
- [4] K. Beck, M. Beedle, A. v. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland und D. Thomas, „Agile Manifesto“, 2001
- [5] CollabNet, VersionOne, „13th Annual State Of Agile Report“, 2019. [Online]  
Available: <https://explore.versionone.com/state-of-agile/13th-annual-state-of-agile-report>
- [6] Scaled Agile, Inc., „Achieving Business Agility with SAFe(R) 5.0“, December 2019. [Online]  
Available: <https://www.scaledagile.com/resources/safe-whitepaper/>
- [7] Scaled Agile, Inc., „Program Increment“, [Online]. Available: <https://www.scaledagileframework.com/program-increment/>
- [8] Scaled Agile, Inc., „PI Planning“, [Online]. Available: <https://www.scaledagileframework.com/pi-planning/>
- [9] Atlassian, „Jira Align Produktseite“, [Online]. Available: <https://www.atlassian.com/de/software/jira/align>
- [10] SoftwarePlant, „BigPicture – Produktwebsite“, [Online]. Available: <https://softwareplant.com/bigpicture/>
- [11] Easy Agile, „Easy Agile Programs – Atlassian Marketplace“, [Online]  
Available: <https://marketplace.atlassian.com/apps/1219491/easy-agile-programs-for-jira>
- [12] Atlassian, „Advanced Roadmaps for Jira“ – Atlassian Marketplace, Atlassian, [Online]  
Available: <https://marketplace.atlassian.com/apps/1221237/advanced-roadmaps-for-jira-in-confluence>
- [13] ALM Works, „Structure – Atlassian Marketplace“, [Online]  
Available: <https://marketplace.atlassian.com/apps/34717/structure-project-management-at-scale>
- [14] catworkx GmbH, „Teamworkx Issue Picker for Jira – Atlassian Marketplace“, [Online]  
Available: <https://marketplace.atlassian.com/apps/1218048/teamworkx-issue-picker-for-jira>

## Rechtliche Anmerkungen

- SAFe® ist eingetragenes Warenzeichen von Scaled Agile, Inc.
- VersionOne CollabNet ist eingetragenes Warenzeichen von CollabNet, Inc.
- VersionOne ist eingetragenes Warenzeichen von VersionOne Inc.
- State of Agile ist eingetragenes Warenzeichen von VersionOne Inc.
- Atlassian, Jira, Jira Software, Jira Align, Confluence sind eingetragene Warenzeichen von Atlassian Inc.

# Über catworkx

**catworkx ist einer der größten Atlassian Platinum und Enterprise Solution Partner in der DACH-Region, mit Standorten in Deutschland (Hamburg, Köln, Stuttgart, München), Österreich (Wien) und in der Schweiz (Winterthur, Luzern), der sich zu 100 Prozent auf Atlassian-Lösungen spezialisiert hat.**

## **Atlassian-Partner der ersten Stunde**

Wir sind Atlassian-Partner und Marketplace-App-Anbieter der ersten Stunde und gehören bereits seit 2002 zu den führenden Dienstleistern im deutschsprachigen Raum. Unser Leistungsangebot, für die gesamte Atlassian-Produktpalette, ist ganzheitlich und umfasst die Prozessberatung und -Implementierung, App-Entwicklung, Plattformlösungen für den Betrieb, offizielle Atlassian-Trainings sowie umfassende Lizenzierungsdienstleistungen. Darüber hinaus ist catworkx Hersteller der Enterprise-Produktfamilie „Teamworkx“.

## **Maßgeschneiderte Services rund ums Atlassian Ecosystem**

Basierend auf intensiver Beratung verfolgen wir bei catworkx einen strukturierten und wertschöpfenden Dienstleistungsansatz und bieten unseren Kunden Atlassian-Lösungen, die oft weit über den Standardeinsatz hinausgehen. Dazu gehören Lösungen für Application Lifecycle Management, IT-Service-Management, Projektportfolio-management oder Lösungen für die Digitalisierung allgemeiner und (skalierter) agiler Geschäftsprozesse.

## **Vom Enterprise-Kunden bis hin zum Mittelstand**

Auch unternehmenskritische Produktinstallationen im Large-Scale-Enterprise-Bereich gehören zu unseren Schwerpunkten, da wir seit über 20 Jahren individuell zugeschnittene Infrastruktur-Lösungen für Großkunden anbieten. Zu unseren Kunden zählen viele renommierte Unternehmen aus nahezu allen Wirtschaftsbereichen wie der Automobilbranche, dem medizinischen Sektor, Banken und Finanzdienstleistungen sowie Logistikunternehmen und Bildungseinrichtungen.

## Kontakt:



**catworkx-Team Deutschland**

Tel.: +49 40 890646-0

[info-de@catworkx.com](mailto:info-de@catworkx.com)



**catworkx-Team Österreich**

Tel.: +43 1 2369317-22

[info-at@catworkx.com](mailto:info-at@catworkx.com)



**catworkx-Team Schweiz**

Tel.: +41 52 56022-20

[info-ch@catworkx.com](mailto:info-ch@catworkx.com)