

## PDM goes SAFe

Einführung eines integrierten, harmonisierten, agilen Projektsetups auf Basis des Scaled Agile Framework (SAFe)

Vortrag auf dem Fachgruppenabend am 8.7.2020

onePDM



Zu meiner Person

# Zu meiner Person

## Frank Brandstetter

- Seit Juli 2019 bei PROSTEP
- Projektleiter für ein zentrales PDM Projekt eines großen Kunden
- Release Train Engineer
- Einer der Treiber der Transformation zu SAFe
- Ehemaliger Management Berater bei Capgemini Invent



## PROSTEP

- Ca. 250 Mitarbeiter
- Aktiengesellschaft
- PLM mit Schwerpunkten:
  - Integrationsprodukte (OpenDXM, Global-X, OpenPDM)
  - PLM Beratung
  - PLM- und Integrations-Projekte

## Das Projekt

- Pflege und Weiterentwicklung eines zentralen, systemwichtigen PDM-Systems des Kunden
- Großer Anteil Legacy-Architektur
- Ca. 30.000 Anwender
- Ca. 70 Projektmitglieder
- Davon 2/3 Sublieferanten
- Läuft bereits seit mehreren Jahren (in unterschiedlichen Konstellationen)

# Agenda

Ausgangsbasis

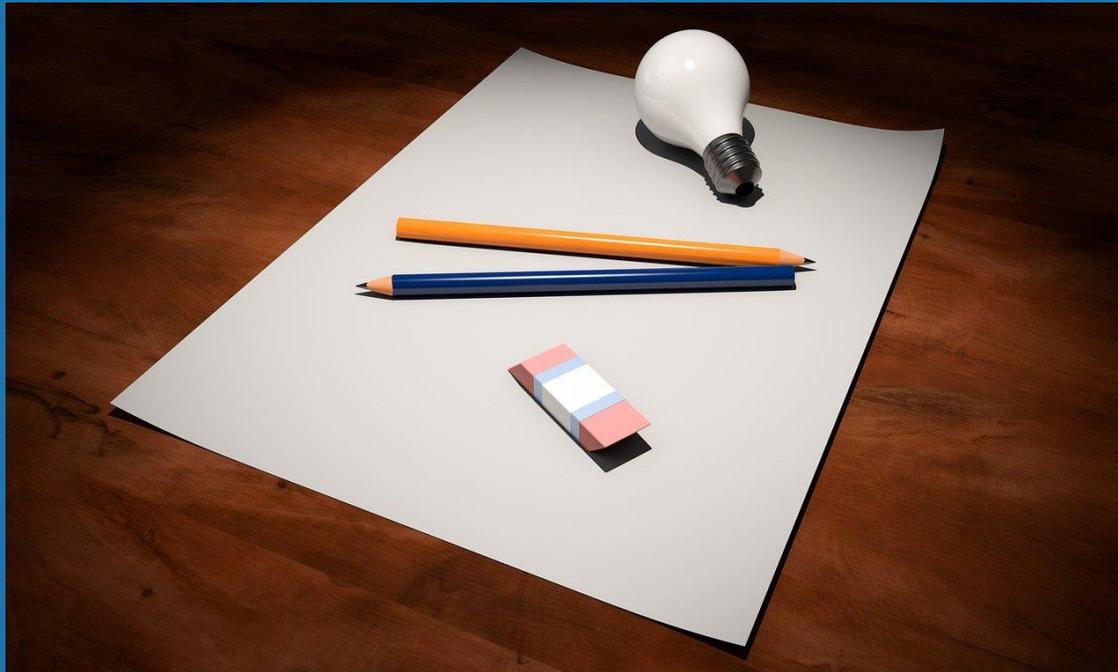
Unser Ansatz

Herausforderungen  
(Gaps)

Die Transformation

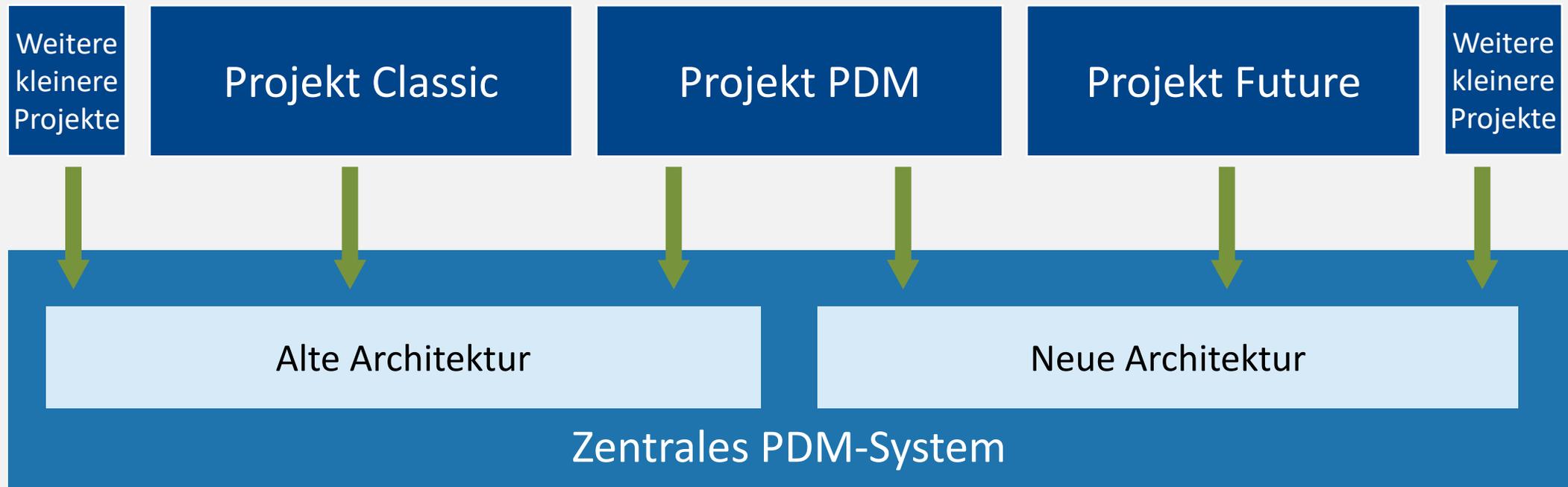
Kritische Betrachtung

Fragen

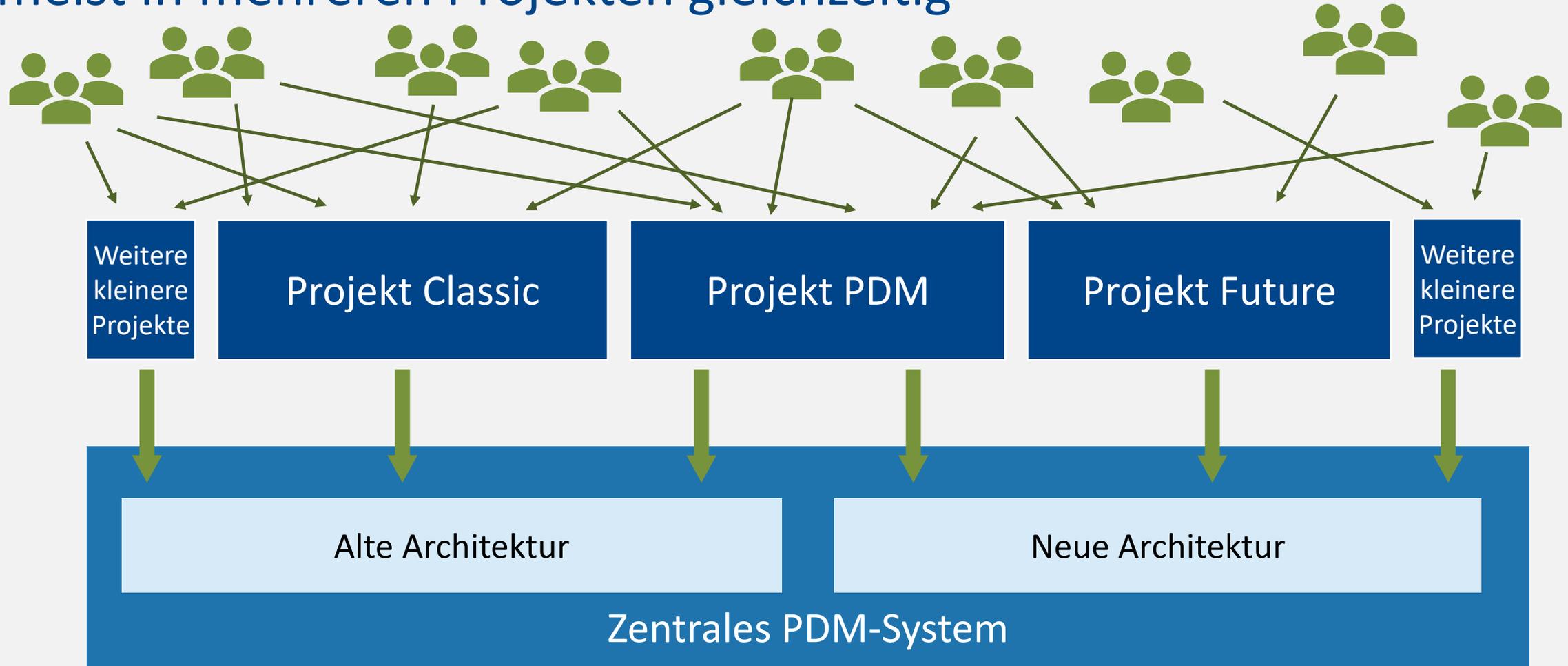


# Ausgangsbasis

# Drei große und mehrere kleine Projekte bestimmten die Situation vor Beginn der Transformation



Es gab sehr viele kleine Teams. Entwickler arbeiteten meist in mehreren Projekten gleichzeitig



## Ressourcenkonflikte

- Mehrere voneinander abhängige Teams
- Die meisten Entwickler arbeiteten in mehreren Projekten gleichzeitig
- Einige Produktowner gingen davon aus, dass ihnen Entwickler direkt zugewiesen sind, auf die sie nach Belieben zugreifen können



## Konsequenzen

- Konflikte durch widersprüchliche Anweisungen an die Entwickler
- Wettkampf um Ressourcen (wer zuerst kommt...)
- Vorrusschauende Ressourcenplanung war schwierig
- Schwer, Zusagen für Meilensteine einzuhalten
- Fehlende Transparenz

# Wissensinseln verhindern Skalierung und bergen das Risiko von plötzlichem Knowhow Verlust

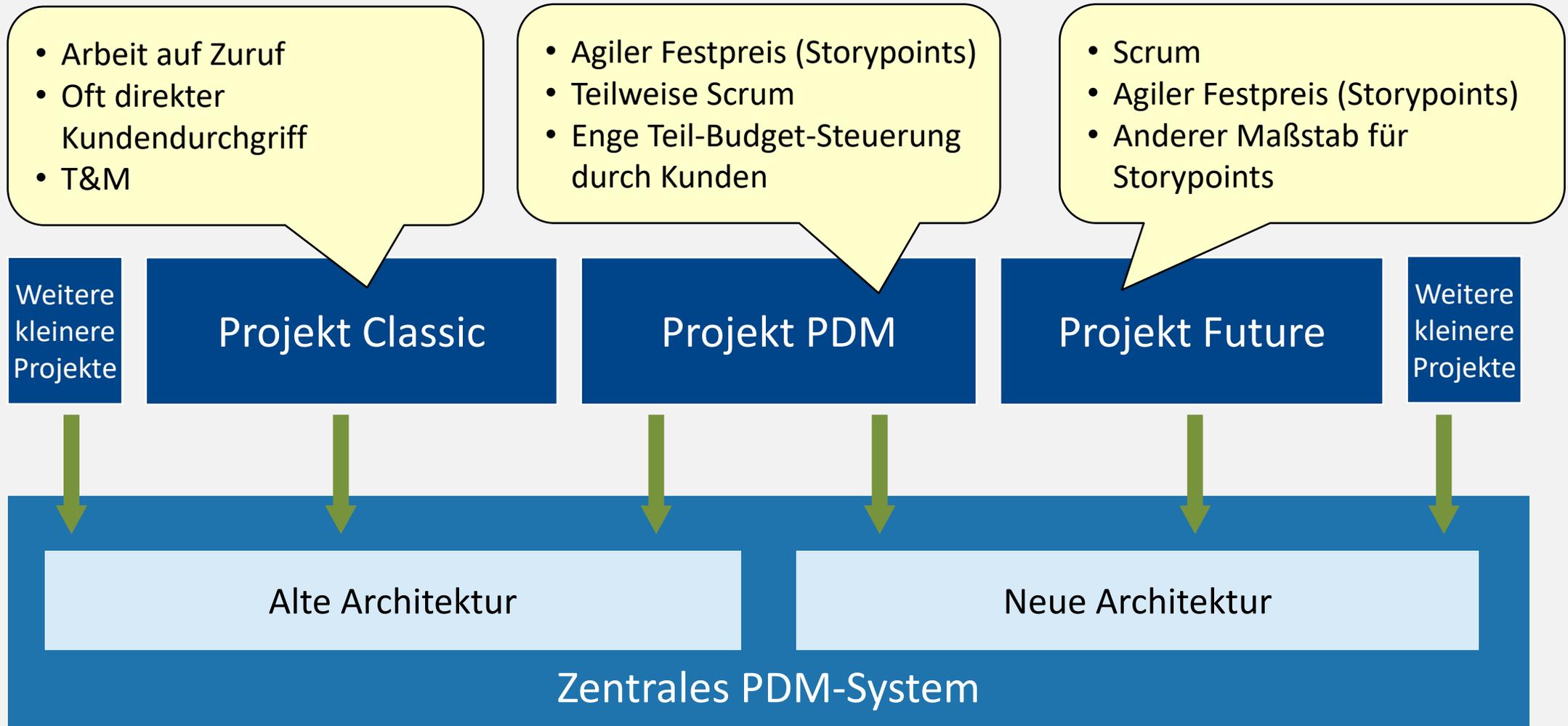
## Konzentriertes Knowhow

- Spezialwissen in wenigen Köpfen konzentriert

## Konsequenzen

- Einzelne Personen sind Engpass-Ressourcen, die das ganze Projekt ausbremsen können
- Skalierung wird erschwert
- Hohes Risiko von plötzlichem Knowhow Verlust durch Fluktuation

# Alle Projekte waren unterschiedlich aufgestellt



# Unterschiedliche Arbeitsmodelle zwischen den Projekten verursachten hohe Koordinationsaufwände und verhinderten Synergien

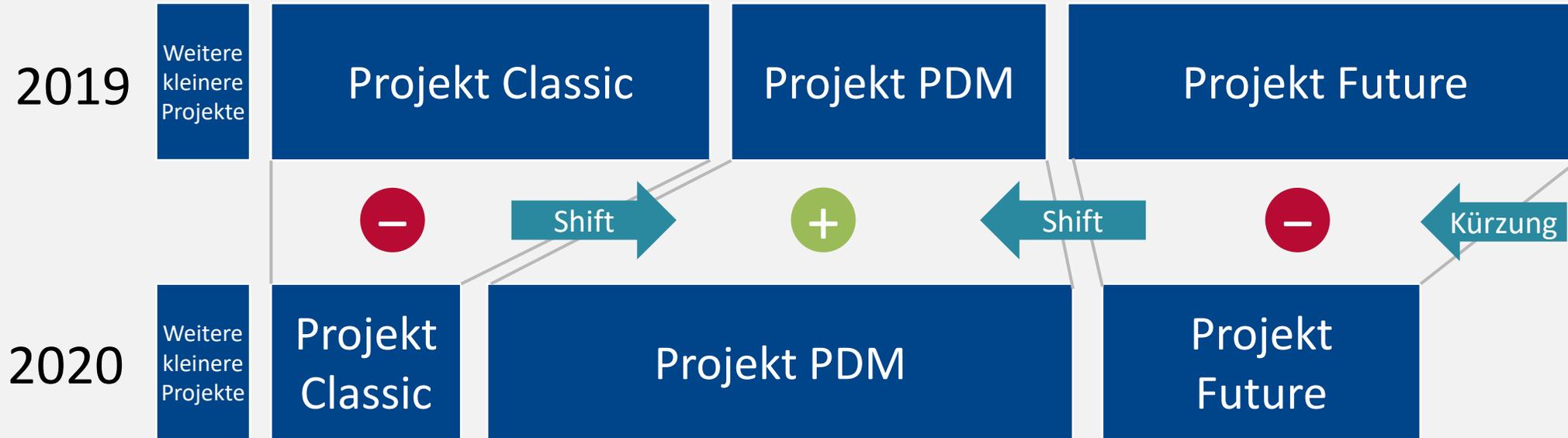
## Unterschiedliche Arbeitsmodelle

- Unterschiedliche Sprintmodelle
- Verschiedene Release- and Deployment-Modelle
- Storypoints mit unterschiedlicher Basis
- Unterschiedliche Abrechnungsmodelle (T&M, Storypoints, Festpreis)
- Unterschiedliche Infrastruktur (SVN, GIT)

## Konsequenzen

- Erhöhte Komplexität
- Hohe Koordinationsaufwände
- Reduzierte Flexibilität
- Verlust von Transparenz

# Zum Jahreswechsel gab es Shifts und Reduktionen der Budgets



Flexibilität und Anpassungsfähigkeit erforderlich



Unser Ansatz:  
PDM goes SAFe

# Das Ziel von **PDM goes SAFe** ist das Zusammenführen der bestehenden PDM Projekte in ein gemeinsames, agiles Projekt

## Ziele

- Zusammenführen der PDM Projekte in ein gemeinsames agiles Projekt
- Einführung gemeinsamer, harmonisierter agiler Prozesse für das zusammengeführte PDM Projekt

## Nutzen

- Bessere Koordination zwischen den PDM-Projekten
- Höhere Transparenz
- Höhere Flexibilität
- Verbessertes Wissensaustausch zwischen den Team Mitgliedern
- Weniger Abhängigkeiten und Engpässe

onePDM

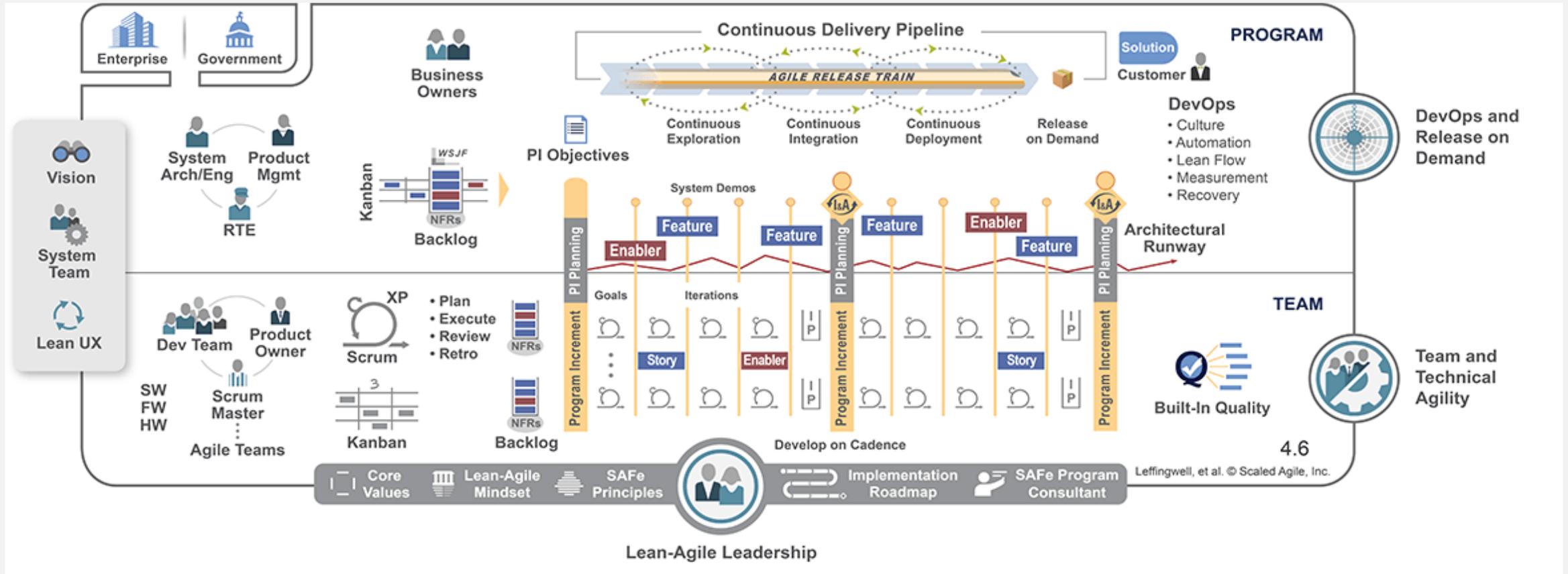
Das neue, zusammengeführte Projekt nennen wir **onePDM**

Das Ziel von **PDM goes SAFe** ist das Zusammenführen der bestehenden PDM Projekte in ein gemeinsames, agiles Projekt

### Scaled Agile Framework (SAFe)

- Da **Scrum** bereits in Teilen eingeführt ist, bot sich Scrum als einheitliches Modell an.
- Als Überbau, zum Koordinieren der Scrum Projekte, wählten wir **SAFe**, da es sich um ein **bewährtes**, in der Praxis **erprobtes Modell** handelt.

# Das Scaled Agile Framework (SAFe) in der Ausprägung „Essential SAFe“ ist die Basis für onePDM

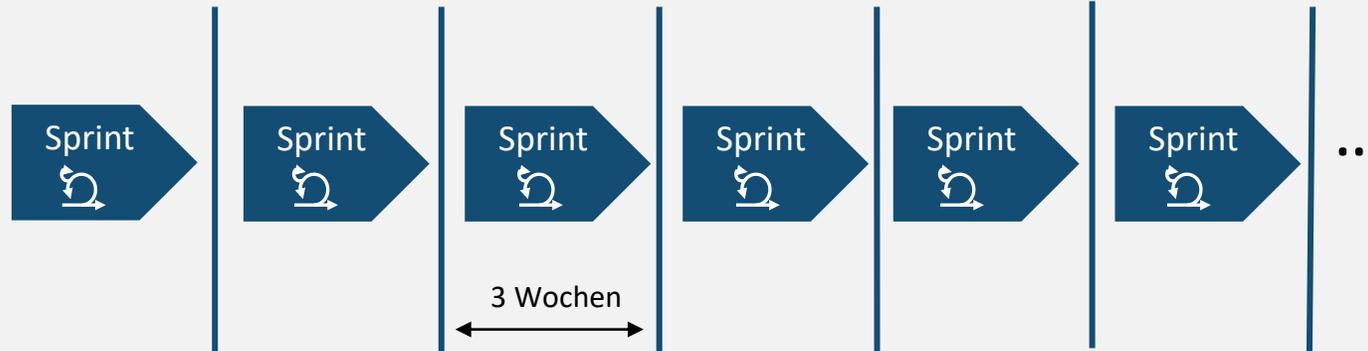


[www.scaledagileframework.com](http://www.scaledagileframework.com)

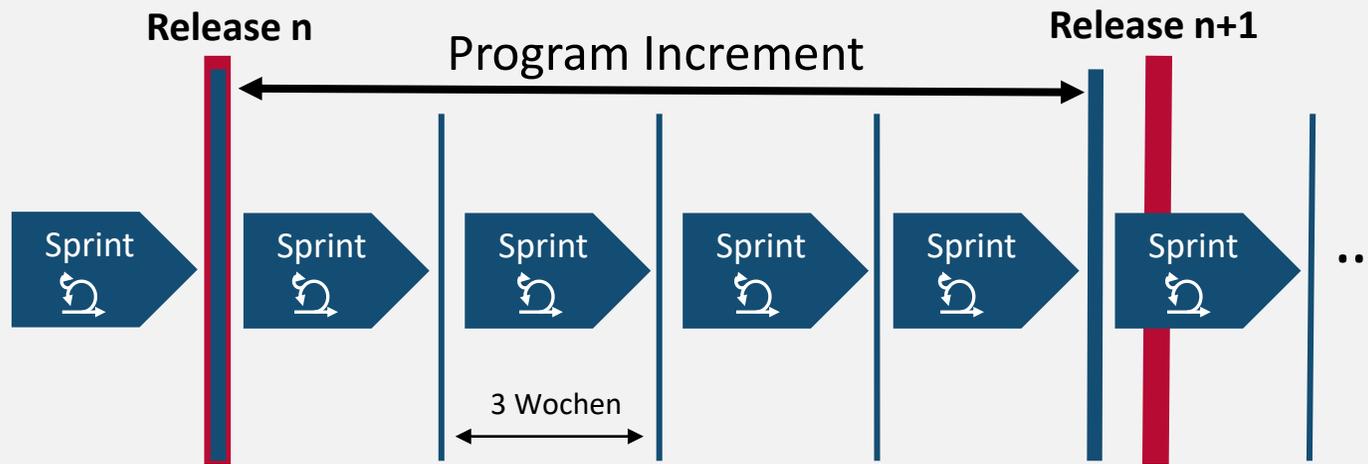
# SAFe ist eine Erweiterung von **standard Scrum**.

Ziel ist eine Skalierung von Scrum auf mehrere Teams

- Jeder Sprint dauert drei Wochen

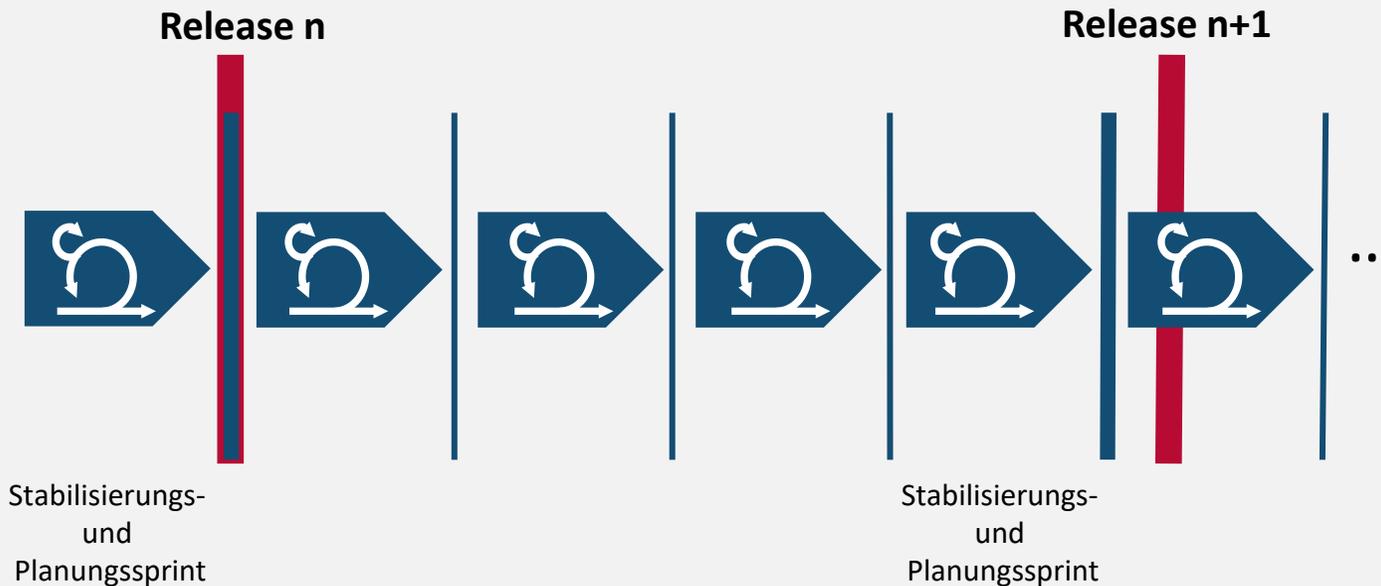


# Bei onePDM entspricht ein **Program Increment (PI)** näherungsweise der Zeit zwischen zwei Releases



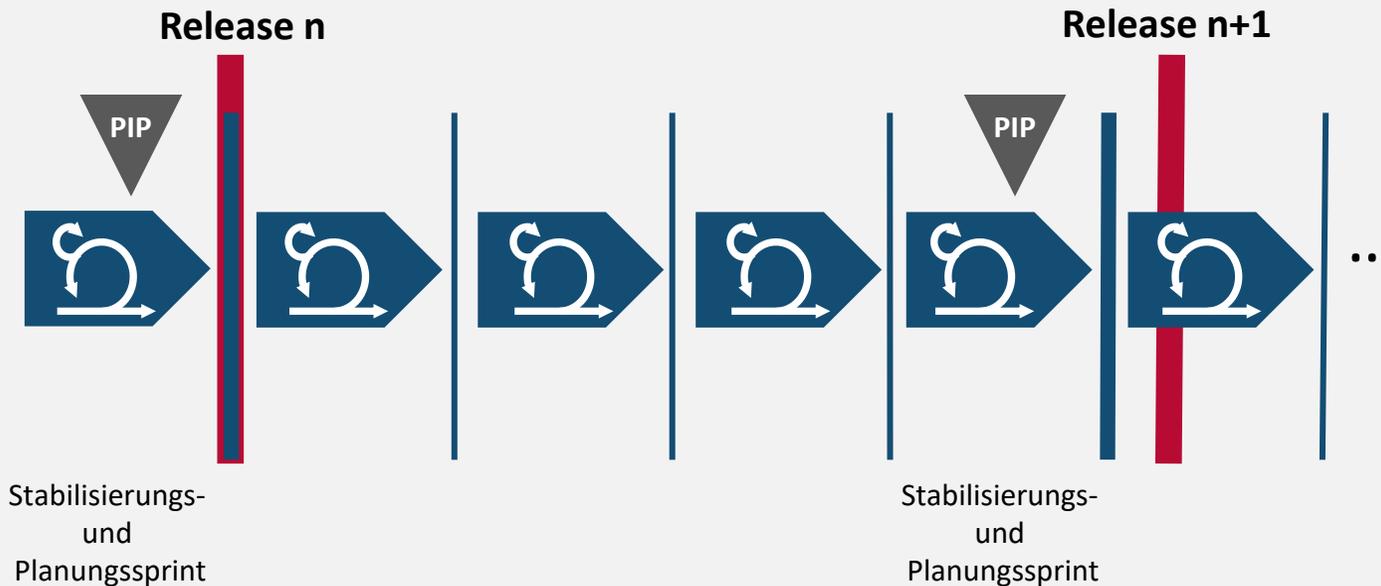
- Jeder Sprint dauert drei Wochen
- Die Anzahl der Sprints pro Program Increment ergibt sich aus der Zeit zwischen den Releases
- Jeder Sprint verläuft nach standard Scrum.

# Der letzte Sprint vor dem Release dient zur **Stabilisierung und Vorbereitung des nächsten PI**



- Den letzten Sprint vor dem PI bezeichnen wir als **Stabilisierungs- und Planungssprint**

# Den Scope für das nächste PI planen wir im Program Increment Planning Meeting (PIP)



- Das PI Planning soll im letzten Sprint vor dem nächsten PI stattfinden

# Die neue Organisation von onePLM hat acht **crossfunktionale agile Teams**

Agiles  
Team 1

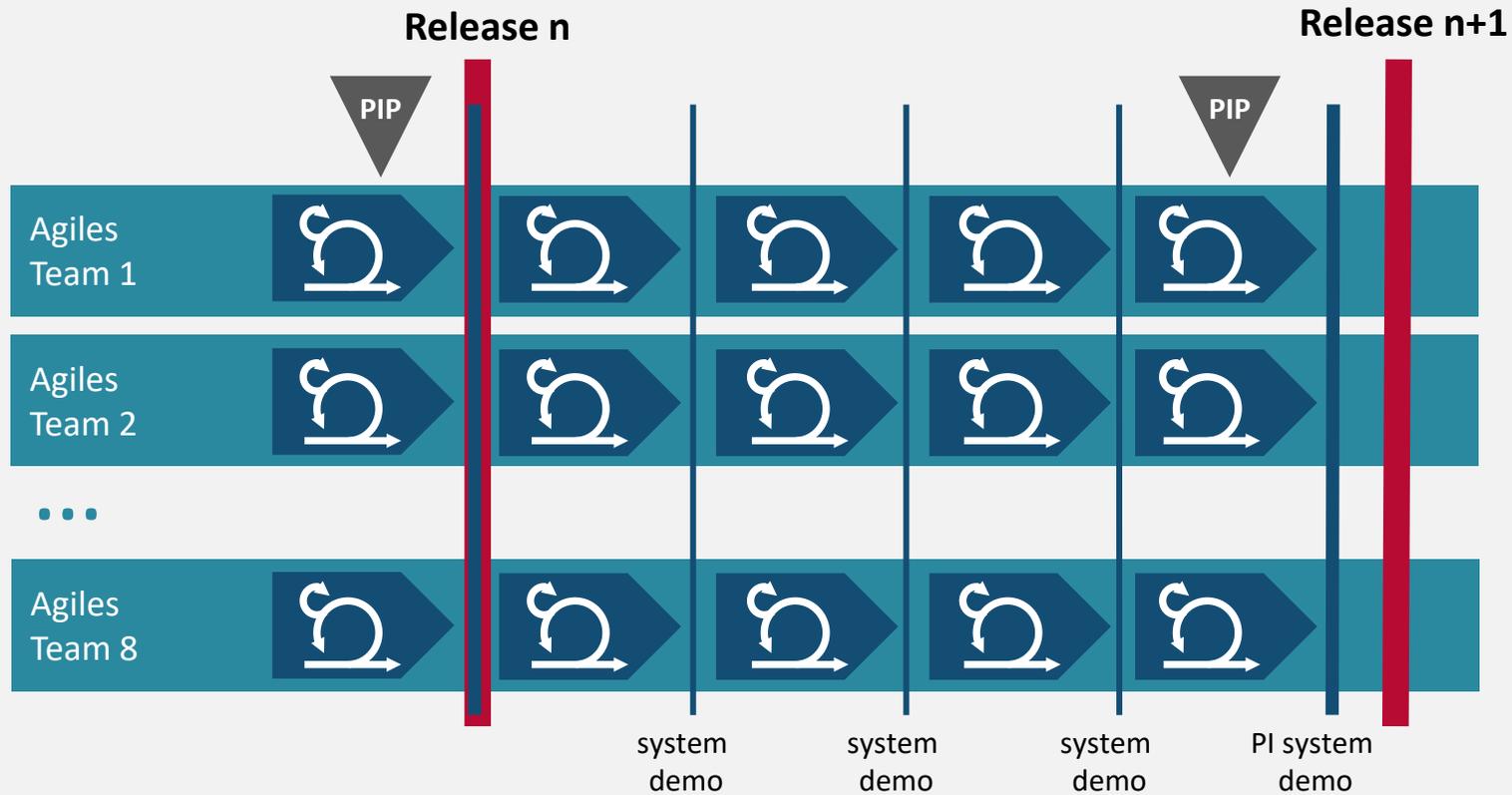
Agiles  
Team 2

...

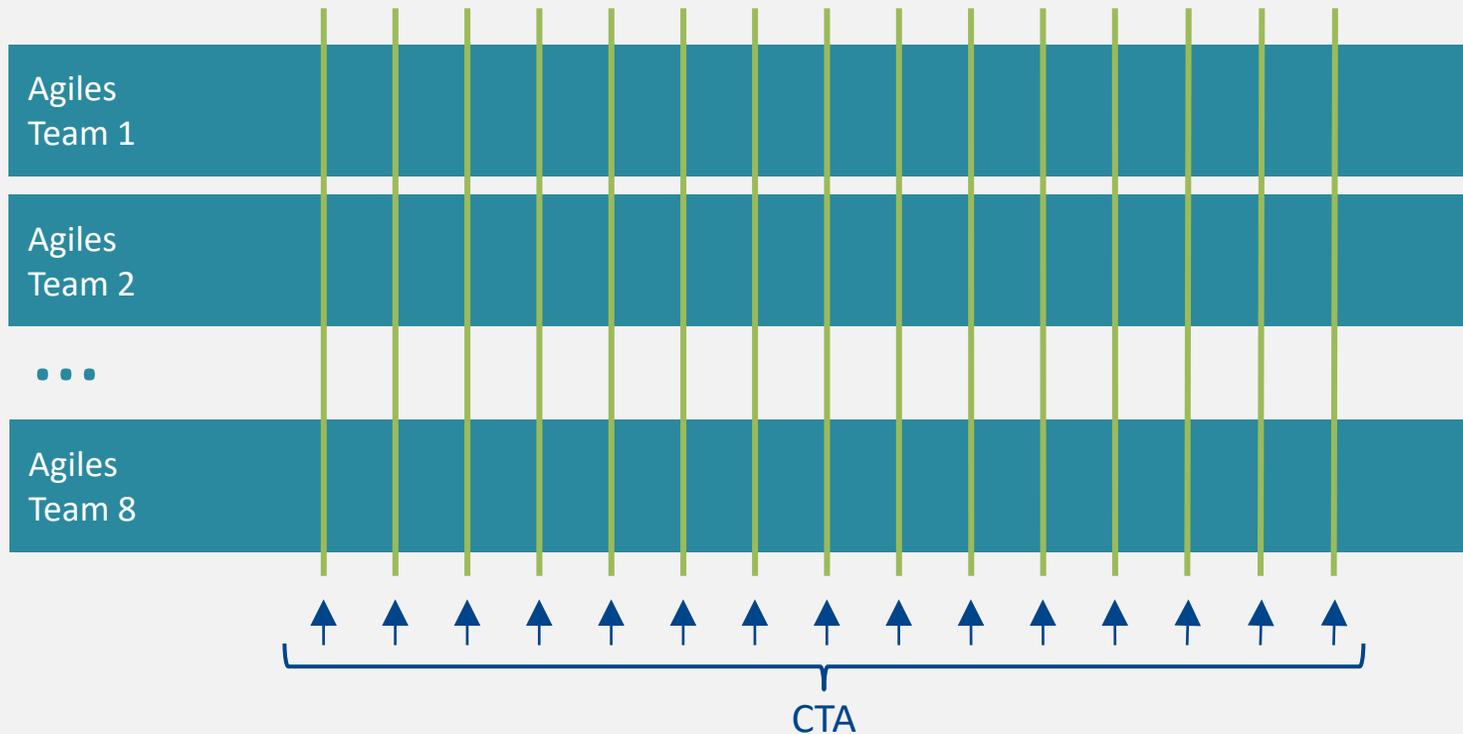
Agiles  
Team 8

- Teams werden nach Standort zusammengestellt
- Teams sollen crossfunktional zusammengestellt sein
- Die derzeitige Spezialisierung der Teams auf architektonische oder technische Themen soll reduziert werden

# Alle Teams arbeiten im **Gleichtakt** mit dem Gleichen Sprint- und PI Timing

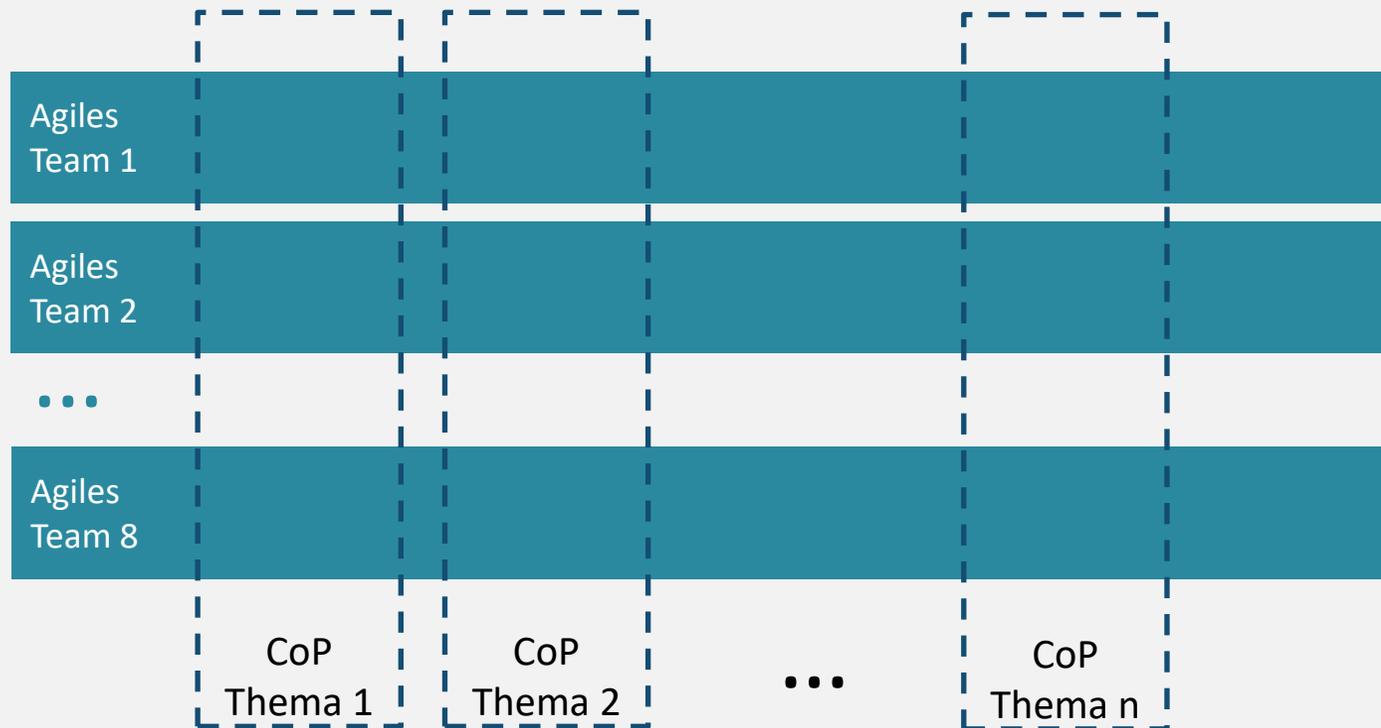


# Ein wöchentliches, leichtgewichtiges **Cross Team Alignment Meeting (CTA)** dient der Teamkoordination



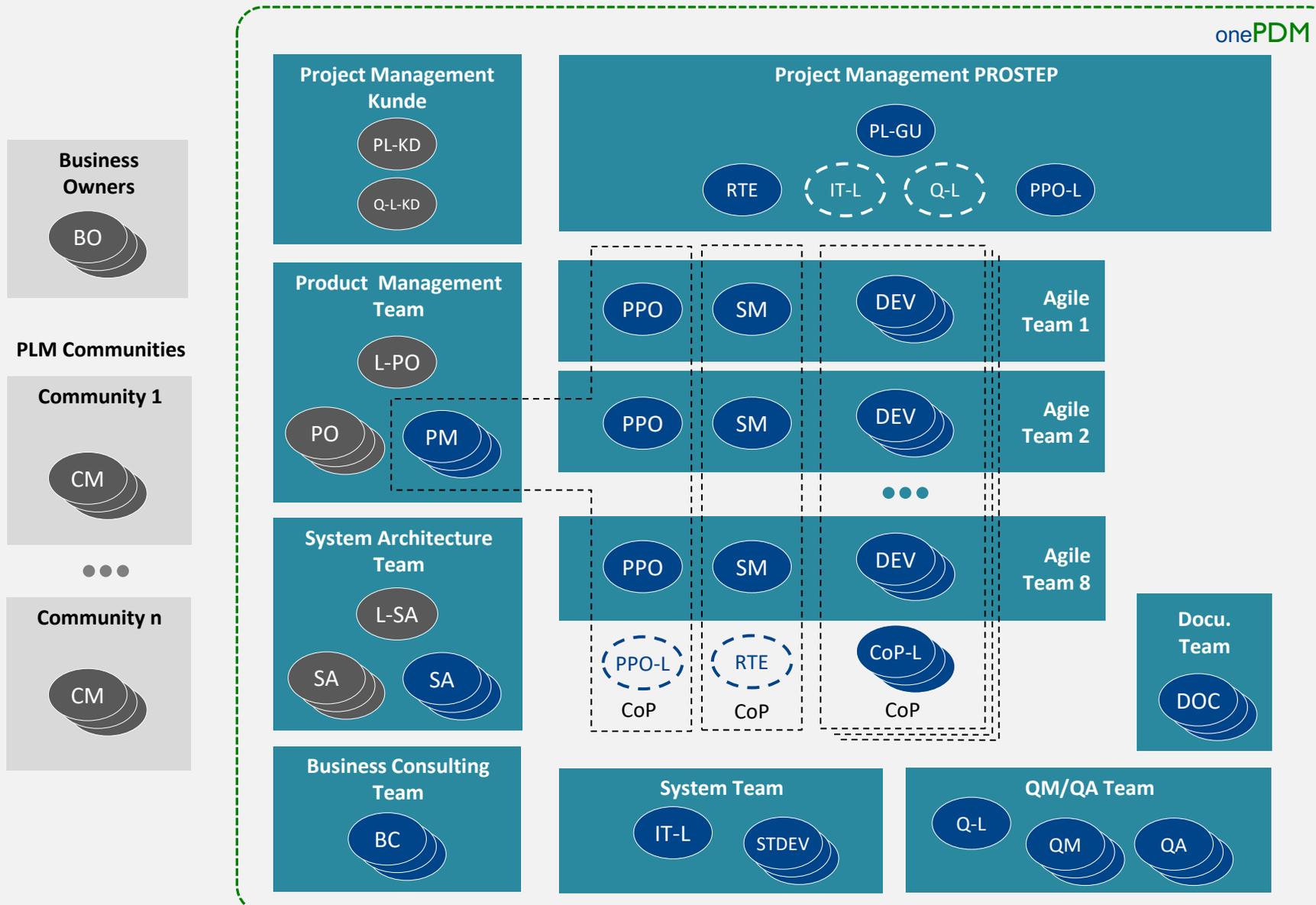
- Jedes Team benennt einen **Team Delegierten** der an dem Meeting teilnimmt
- Das Cross Team Alignment Meeting wird ähnlich wie ein Daily Scrum durchgeführt

# Um den Wissenstransfer zwischen den Teams zu fördern, führen wir **Communities of Practice (CoP)** ein



- **Communities of Practice** sind eine Plattform für Teammitglieder mit einem gemeinsamen Fokus zum Austausch von Wissen und für gegenseitigem Rat
- Ein **CoP-Leiter** moderiert die CoP und fördert die Kommunikation und den Wissensaustausch

# Die neue Projektorganisation orientiert sich an SAFe **PROSTEP**



onePDM

## Project Roles:

- BC Business Consultant
- CoP-L Community of Practice Lead
- DEV Developer
- DOC Document. Team Member
- IT-L IT-Lead
- L-PO Lead Product Owner
- L-SA Lead System Architect
- PL-GU Project Manager PROSTEP
- PL-KD Project Manager Kunde
- PM Product Manager
- PO Product Owner
- PPO Proxy Product Owner
- PPO-L Proxy Product Owner Lead
- Q-L Quality-Lead
- QA Quality Assurance / Tester
- QM Quality Manager
- RTE Release Train Engineer
- SA System Architect
- SM Scrum Master
- STDEV System Team Developer

## Other Roles:

- BO Business Owner

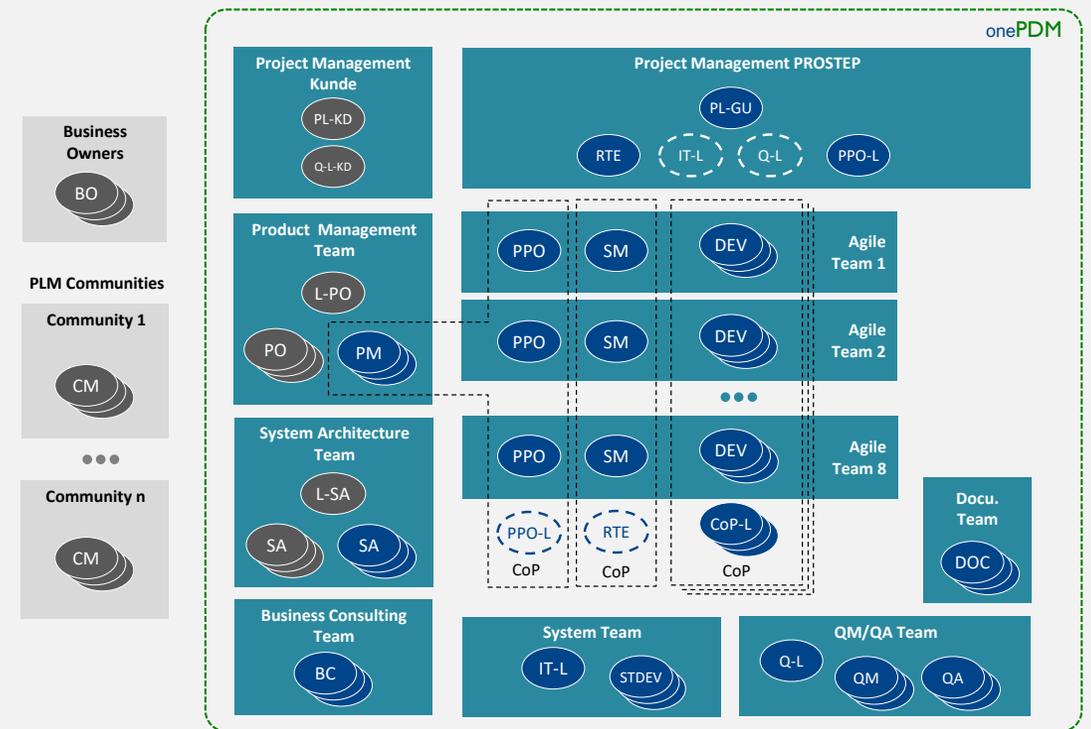
## Abbreviations:

- CoP Community of Practice

## Legend:



# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



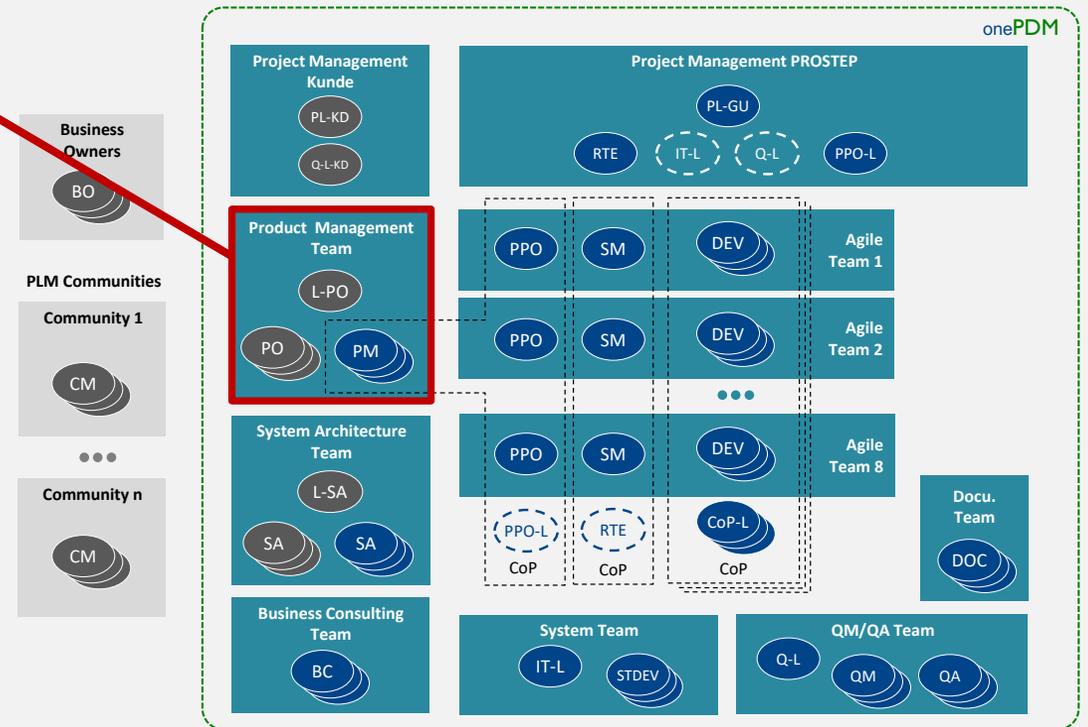
# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



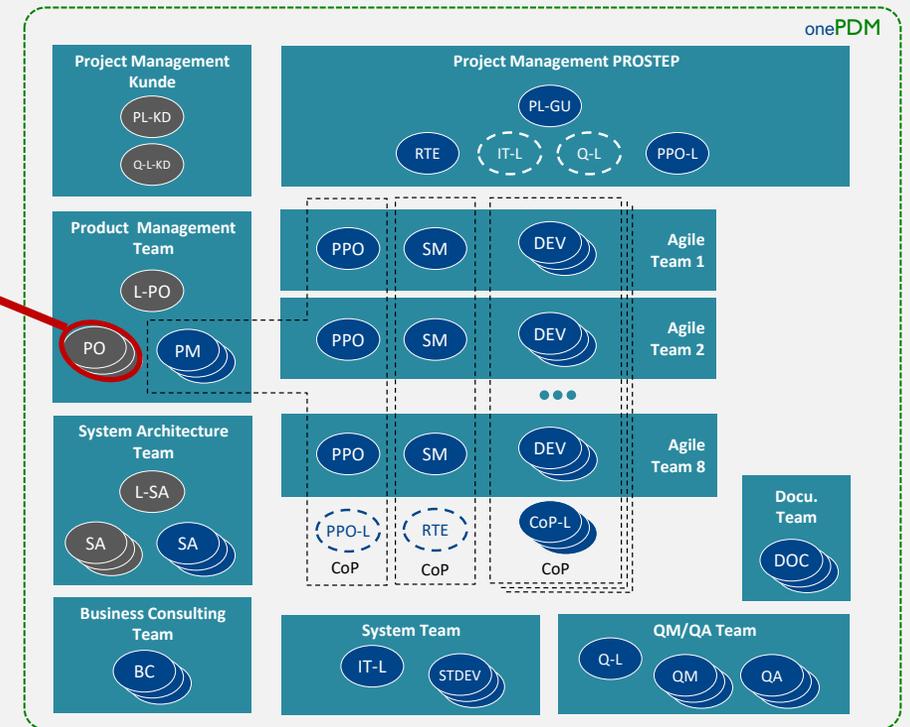
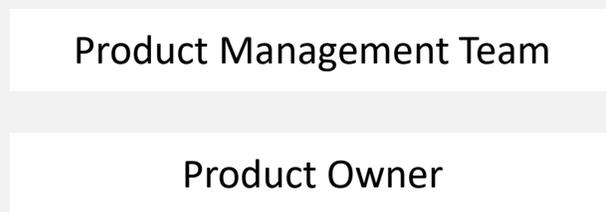
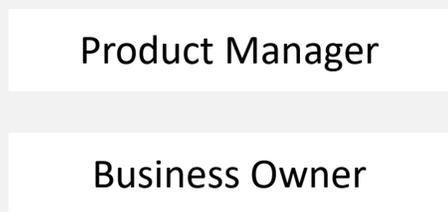
Product Manager



Product Management Team



# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab

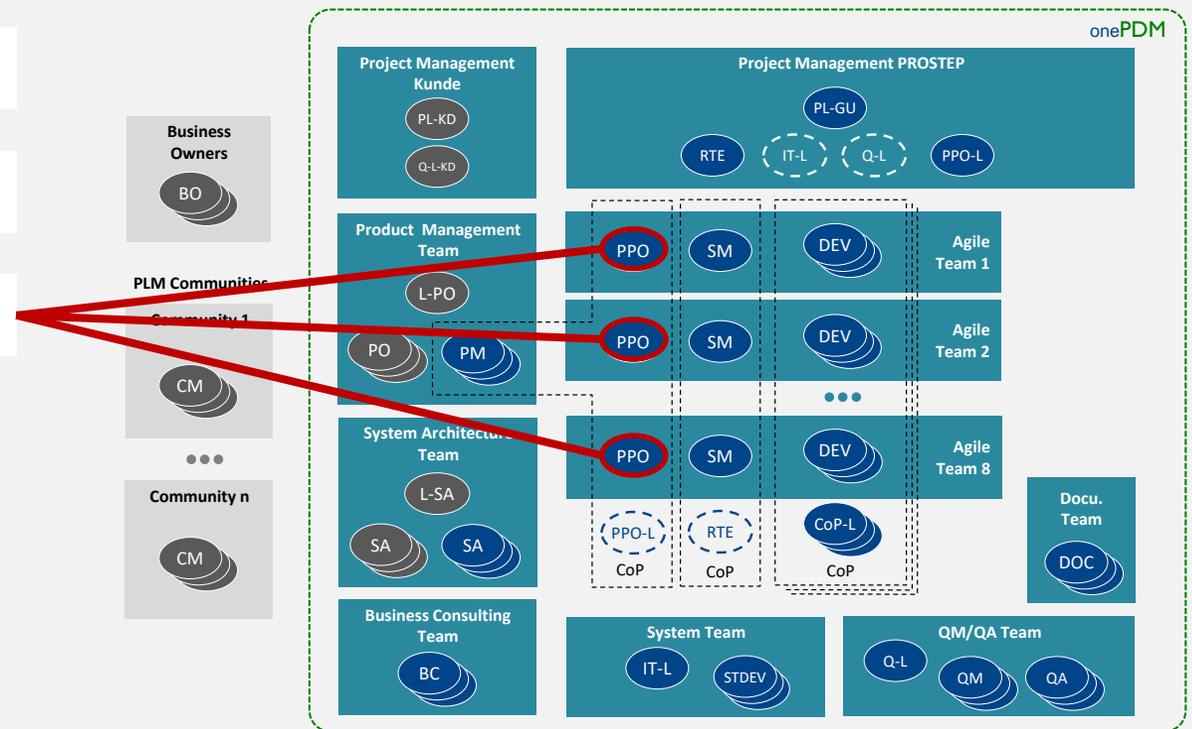


onePDM

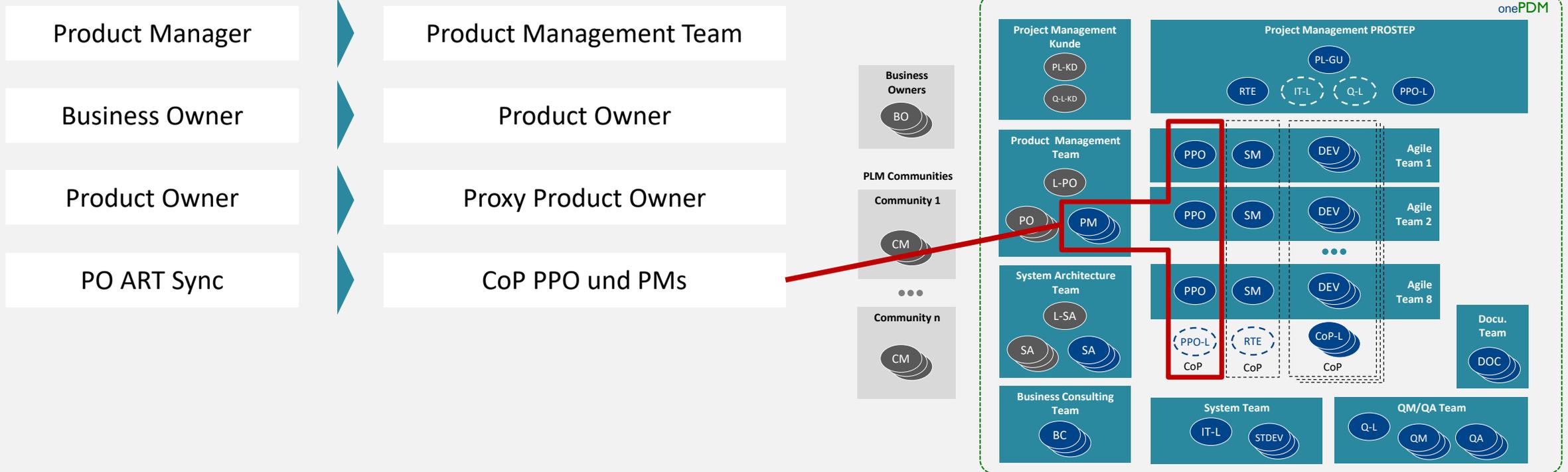
- Product Manager
- Business Owner
- Product Owner



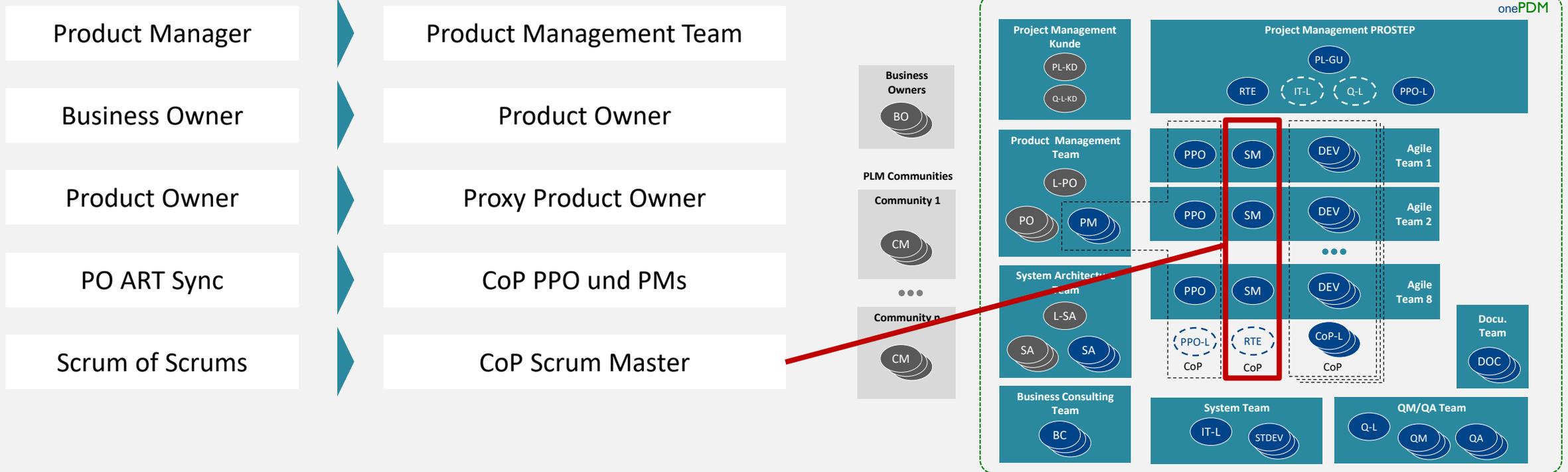
- Product Management Team
- Product Owner
- Proxy Product Owner



# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



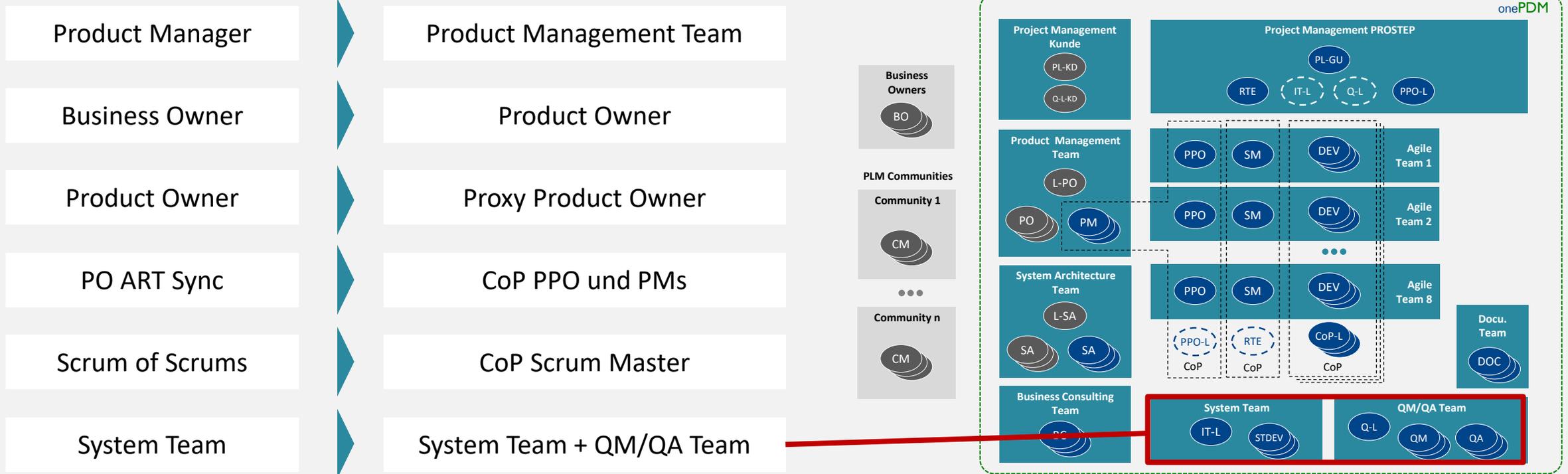
# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



# Aus historischen und politischen Gründen weichen manche Rollenbezeichnungen von denen in SAFe ab



onePDM

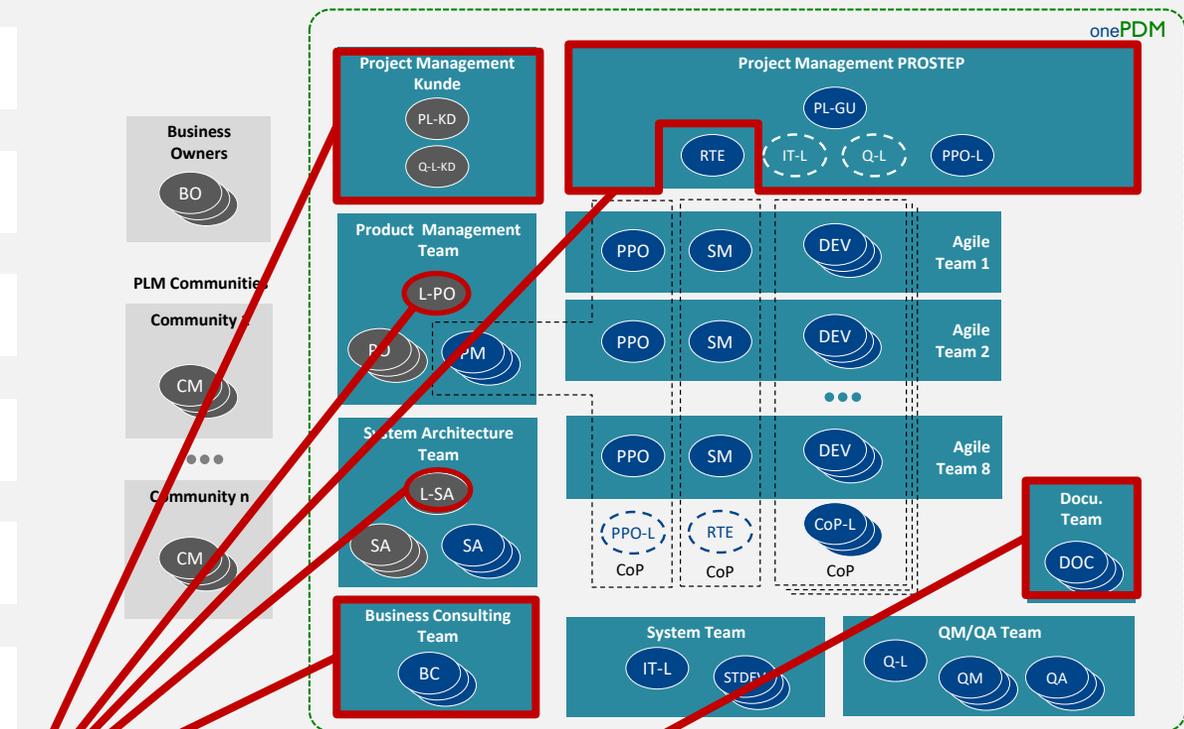


# Es gibt außerdem gegenüber SAFe zusätzliche Teams und Rollen



onePDM

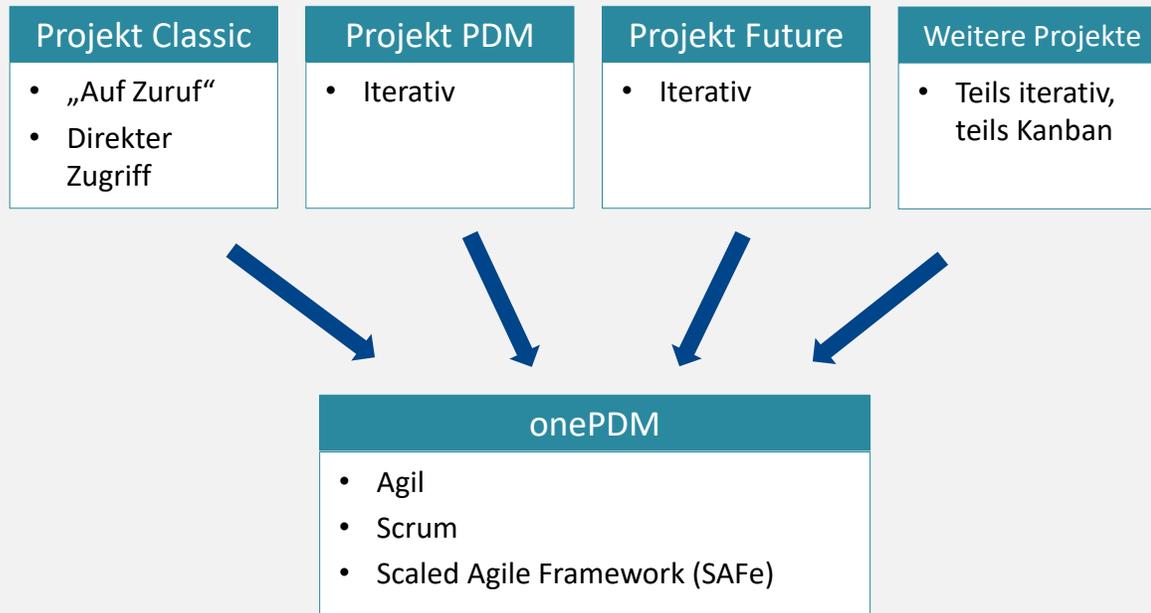
Product Manager	Product Management Team
Business Owner	Product Owner
Product Owner	Proxy Product Owner
PO ART Sync	CoP PPO und PMs
Scrum of Scrums	CoP Scrum Master
System Team	System Team + QM/QA Team
	Weitere (nicht SAFe) Teams und Rollen





## Herausforderungen (Gaps)

# Unterschiedliche Entwicklungsprozesse



## Situation

- Alle Projekte folgten unterschiedlichen Entwicklungsmodellen
- Entwickler mussten sich beim Wechsel zwischen den Projekten auf andere Entwicklungsprozesse umstellen

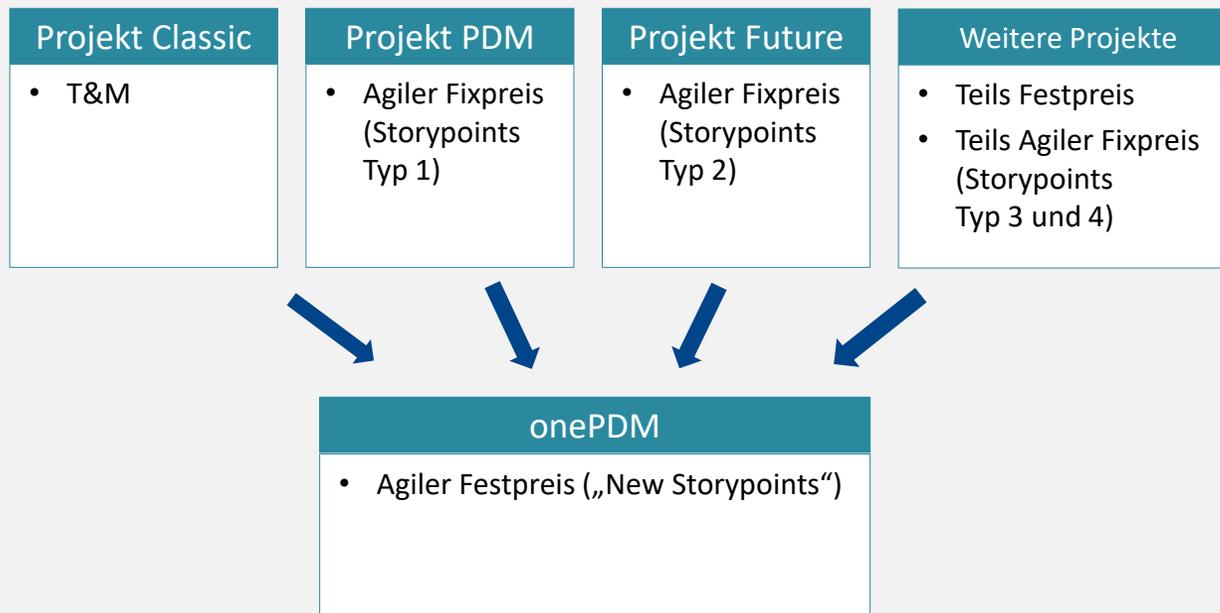
## Transformation

- Da **Scrum bereits in Teilen eingeführt** ist, bot sich Scrum als einheitliches Modell an.
- Als Überbau, zum Koordinieren der Scrum Projekte, wählten wir **SAFe**, da es sich um ein **bewährtes**, in der Praxis **erprobtes Modell** handelt.
- Manche Entwickler hatten bereits **Erfahrung mit Scrum**, andere mussten noch Erfahrungen mit der agilen Vorgehensweise sammeln
- Das Framework SAFe war praktisch **allen neu**.

### Methode:

- Trainings & Coachings

# Unterschiedliche Abrechnungsmodelle, unterschiedliche Schätzmaßstäbe



## Situation

- Viele verschiedene Abrechnungsmodelle
- Unterschiedliche Schätzmaßstäbe für Storypoints
- Erfassung und Abrechnung erschwert
- Reduzierte Flexibilität
- Kunde äußerte Wunsch auf Vereinheitlichung

## Transformation

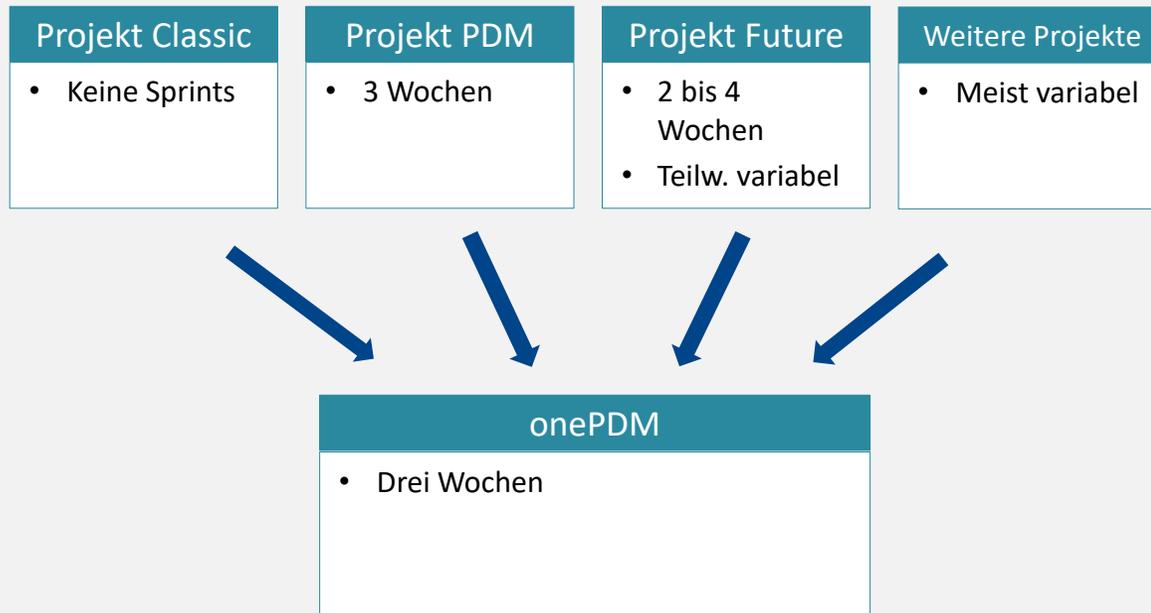
### Herausforderung:

- die Umstellung der Abrechnungsmodelle, so dass am Ende **für alle** möglichst **das Gleiche wie vorher** herauskommt

### Methode:

- Abrechnung über **Storypoints**
- Einführung von „**New Storypoints**“.
- T&M bzw. Festpreise wurden so in „New Storypoints“ **umgerechnet**, dass am Ende im Mittel die gleiche Leistung fürs Geld herauskommt
- Hier hat sicherlich geholfen, dass der Kunde die Vereinheitlichung auch wollte.

# Unterschiedliche Sprintzyklen



## Situation

- Unterschiedliche Dauer der Iterationen je nach Projekt und Team

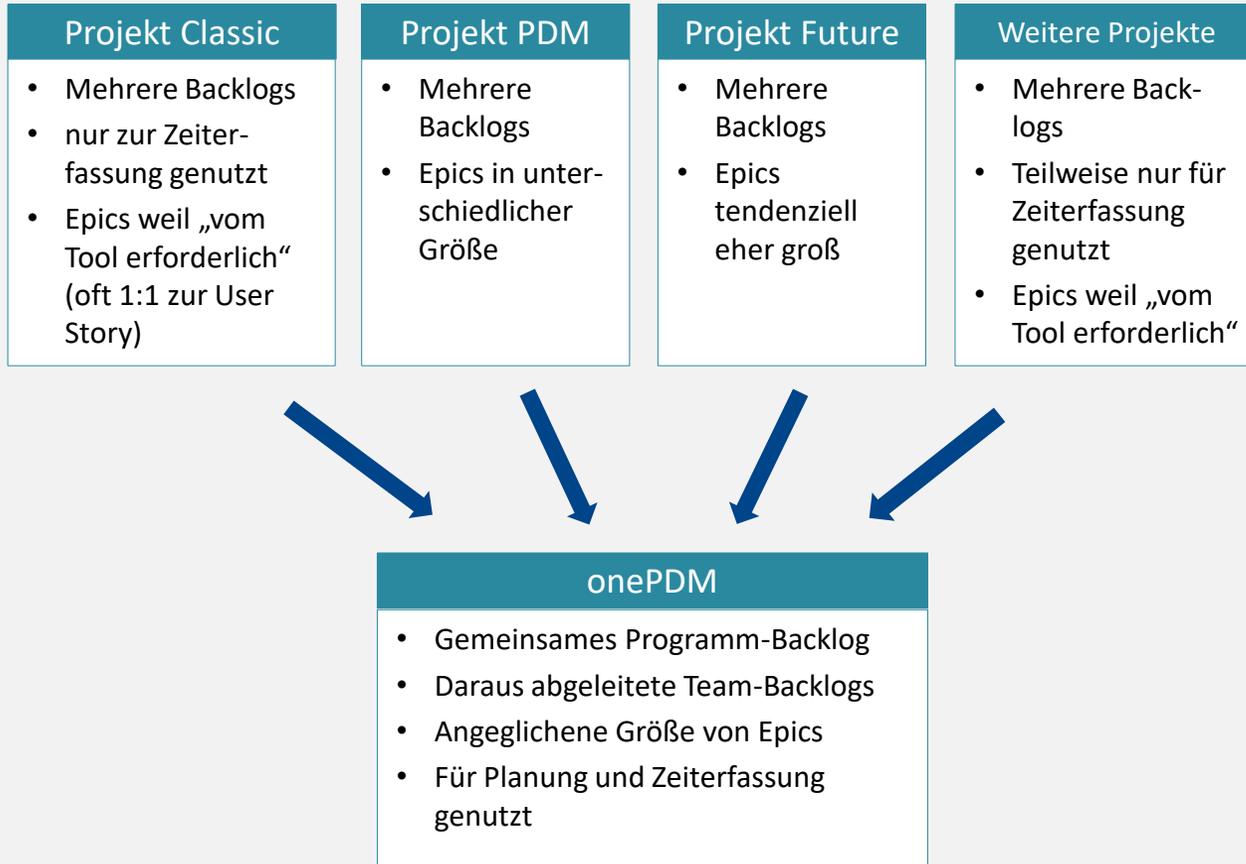
## Transformation

- Dreiwöchige Sprints
- War im Projekt PDM bereits etabliert
- Geeigneter Kompromiss für alle
- Overhead auf angemessenes Maß reduziert

### Methode:

- Abstimmung und **individuelle Regelungen** zur Sprintanpassung

# Zusammenführen der Backlogs



### Situation

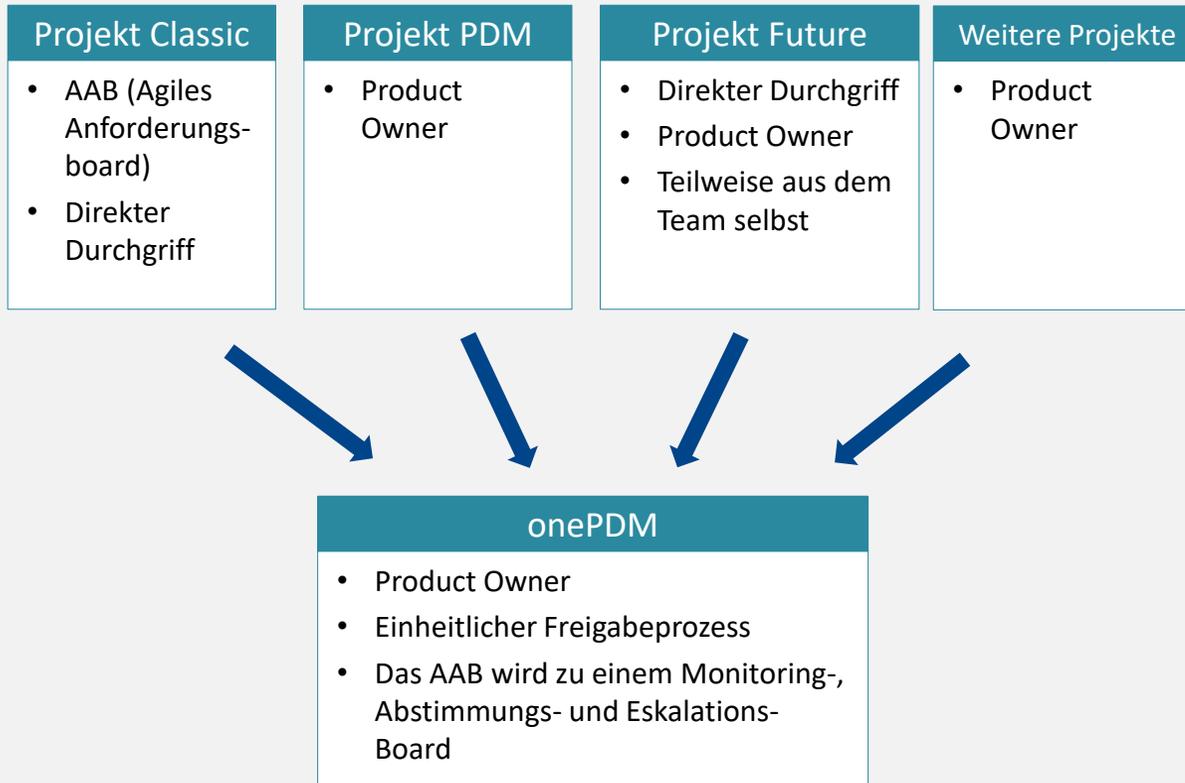
- Backlogs wurden auf unterschiedlicher Basis gepflegt
- Epics haben unterschiedliche Granularitäten

### Transformation

**Vorgehen:**

- Ein neues Element „**Big Point**“ (in SAFe wäre das ein Epic) wurde eingeführt
- Die neue Strukturierung wurde **kommuniziert**
- Die verschiedenen, bisher getrennten Backlogs wurden in ein neues, **gemeinsames Backlog** verschoben
- Überführung in neue Struktur und Angleichung der Granularität erfolgte **sukzessive** durch PO, PM und PPOs

# Unterschiedliche Anforderungsprozesse



## Situation

- Die Anforderungsprozesse waren zwischen den Projekten unterschiedlich
- Das Agile Anforderungsboard (AAB) war ein Engpass
- Anforderungen wurden teils unabgestimmt ins Team getragen

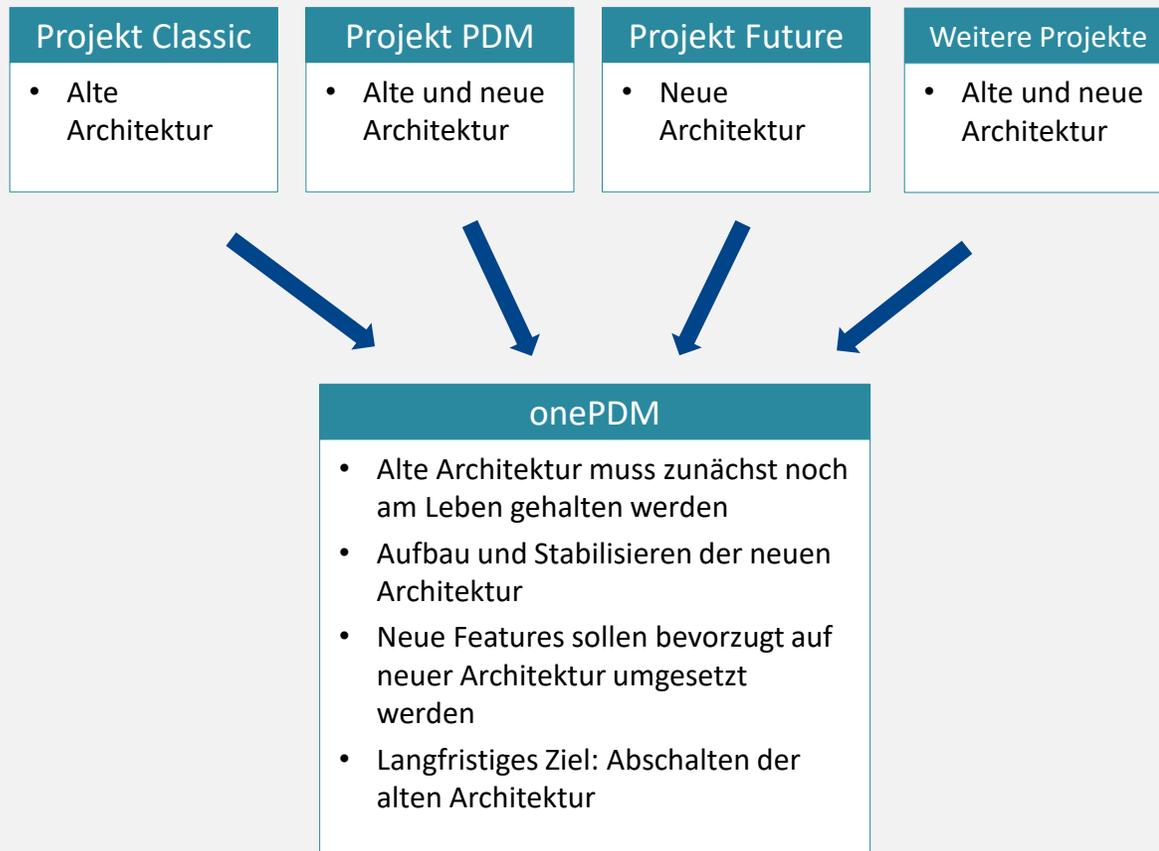
## Transformation

- Einführung eines einheitlichen Anforderungsprozesses auf Basis SAFe
- Das AAB wandelt sich von einem Freigabeboard zu einem Koordinations- und Eskalationsboard
- Zweistufige Priorisierung:
  1. Durch Budgettöpfe
  2. Innerhalb der Budgettöpfe durch die verantwortlichen Product Owner

### Methode:

- Viele Erläuterungs- und Abstimmungsrunden
- Hat im Wesentlichen der Kunde gemacht

# Architektur-Shift und die Struktur der Teams



## Situation

- onePDM baut im wesentlichen auf zwei verschiedenen Architekturen auf: der legacy Architektur und der neuen Future Architektur.
- Beide Architekturen wurden in der Vergangenheit im Wesentlichen von unterschiedlichen Teams gepflegt.

## Transformation

- Wissen über die unterschiedlichen Architekturen muss in den Teams zusammengeführt werden.
- D.h. die Teams sollten durchmischt werden

### Herausforderung:

- derzeit bestehende (Technologie-) Teams nicht völlig auseinanderreißen
- gleichzeitig für Wissenstransfer durch eine Durchmischung sorgen

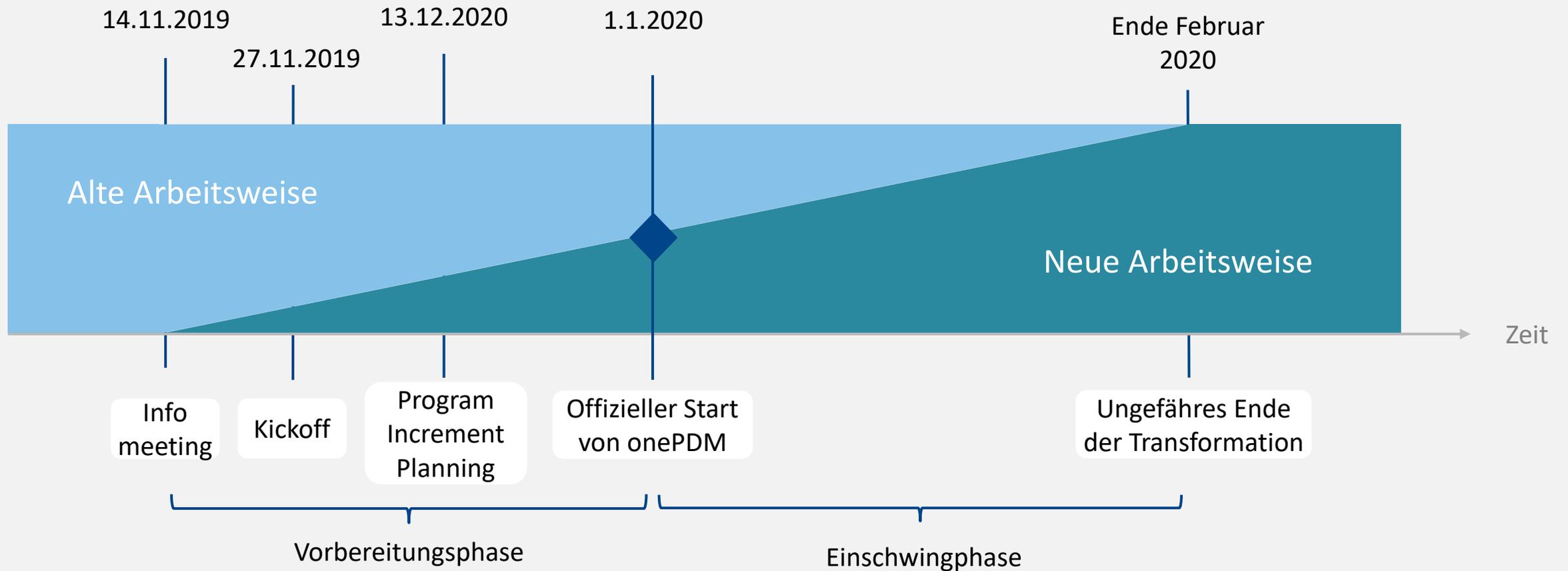
### Methode

- Teams so schneiden, dass Features möglichst in einem Team umgesetzt werden können (Abhängigkeiten minimieren)
- Wissensträger zu einzelnen Technologien soweit möglich auf verschiedene Teams verteilen



# Transformation

# onePDM startete offiziell am 1.1.2020





Kritische Betrachtung

## Was haben wir erreicht?



- Großer Schritt vorwärts in Richtung **Vereinheitlichung**
  - Entwicklungsprozesse
  - Anforderungsprozess
  - Abrechnung / Schätzung
  - Gemeinsames Backlog
- Klar definierte **Projektstruktur**
- Klar definierte **Rollen**
- Klar definierte **Kommunikationswege**
- Aktuelle **Feature Teams** performen gut
- Höhere **Transparenz**
- Höhere **Flexibilität**

## Wo können wir noch verbessern?



- Beim Thema **Wissenstransfer** sind wir noch nicht am Ziel:
  - Es gibt immer noch Engpass-Ressourcen
  - Wissen über Legacy- und Neue-Architektur ist immer noch weitgehend getrennt
- **Agiles Verständnis** ist noch nicht in allen Teams angekommen
- **Verbindlichkeit** ist besser geworden, aber noch nicht gut
- **Zu viele Product Owner** auf Seiten des Kunden behindern Entscheidungen (spezielles Problem des Kunden)
- **Informationsfluss** auf **Kundenseite** verbessern
- Mangelnde Bereitschaft, für **SAFe Geld** auszugeben (insbesondere für Großveranstaltungen wie PI Planning und Inspect & Adapt)
- **Communities of Practice** könnten aktiver sein

## Was sollten wir wieder so machen?



- Kein Big Bang, sondern eine **Transformationsphase**
- Neue Teamstrukturen frühzeitig kommunizieren
- Der Ansatz, erst einmal **alle Teams neu zusammenzustellen** hat für die notwendige **Aufbruchstimmung** gesorgt.
  - Führte aber auch zu einem gut zu einem beobachtbaren Durchlauf des Team-Zyklus (Forming, Storming, Norming, Performing)
- Mehrere, ausführliche **Infomeetings** mit Fragestunde
- CoP Scrum Master (**Scrum of Scrums**)
- CoP PPO/PM (**PO ART Sync**)

## Was sollten wir beim nächsten Mal anders machen?



- Der erste Ansatz **rein crossfunktionaler Teams** ist über das Ziel hinaus geschossen. Lieber gleich auf Feature Teams setzen.
- **Programm Incremente** sollten **kürzer** werden. Pls müssen nicht mit Releases zusammenfallen.
- Auf **Kundenseite** den **SAFe Prozess** frühzeitiger mit allen wichtigen **Stakeholdern** abstimmen.
- Noch **mehr agile Schulungen**. Nicht auf Schulungen aus der Vergangenheit vertrauen.
- Die aus politischen und historischen Gründen eingegangenen **Abweichungen** gegenüber der **SAFe Terminologie** führen zu Verwirrungen
- Anfangs gab es zu **wenig Kommunikation** über Prozessänderungen und Beschlüsse **ins Team**
- Ein **agiler Festpreis** macht mehr Probleme als er löst! Insbesondere das Schätzen wird zum Issue!

SAFe ist ein guter Leitfaden für **onePDM**.  
Wir werden uns weiterhin daran orientieren.

**onePDM** ist eine **angepasste Projektmethodik**, die Konzepte aus SAFe mit den speziellen Anforderungen des Kunden vereint.



Fragen?



Vielen Dank