



Fachhochschule Nordwestschweiz  
Hochschule für Technik

## SOFTWARE-ENTWICKLUNGSPROZESSE

### Aufgabe 2.2

## *Gegenüberstellung Scrum - DAD*

**13. November 2019**

**Dozentin:**

*Peter Sybille*

**Studierende:**

*Eggenschwiler Carlo*

*Frei Dominik*

*Frommenwiler Dominic*

*Inniger Marco*

# Zusammenfassung

Diese Arbeit liefert eine Gegenüberstellung von Scrum und Disciplined Agile Delivery. Der Fokus wird auf die Themen Planung, Zusammenarbeit und Zuständigkeit gelegt. Es werden Grundkenntnisse über das Vorgehensmodell Scrum oder Extrem Programming (XP) vorausgesetzt.

Disciplined Agile Delivery (DAD) basiert grösstenteils auf Scrum und Extreme Programming. Im Zentrum steht die Construction Phase mit Iterationen (Sprints) von fixer Zeitdauer. Dieses Vorgehensmodell findet dann Anwendung, wenn Scrum den Bedürfnissen nicht mehr genügt und erweitert werden muss und ein agiles Projekt durchgeführt werden muss.

DAD ist **iterationsbasiert**. Wie bei vielen agilen Methoden, einschliesslich Scrum und XP, wird die Lösung schrittweise und zeitgesteuert aufgebaut. Diese Zeitrahmen werden Iterationen genannt (was Scrum Sprints nennt).

DAD verwendet **nicht die Scrum Terminologie**. Die Entwickler von DAD haben sich explizit gegen die Scrum Terminologie entschieden. Jedoch spielt diese keine Rolle. So darf in DAD mit der Scrum Terminologie gearbeitet werden. Im Blog «Disciplined Agile Terminology» [1] wird erklärt warum das Team um Scott Ambler sich gegen die Scrum Terminologie entschieden hat.

DAD zeigt **Eingaben von ausserhalb des Lieferlebenszyklus** auf. DAD zeigt, dass vor Beginn des Projekts etwas passiert und dass agile Teams oft neue Anforderungen (in Form von Änderungsanfragen und Fehlermeldungen) aus der Produktion erhalten. Diese Inputs liefern einen wichtigen Kontext für den gesamten Lieferlebenszyklus.

Es gibt eine **Workitem-Liste** und kein Product Backlog. DAD hat einen grösseren Umfang als Scrum, und wenn man diesen grösseren Umfang berücksichtigt, beginnt man zu erkennen, dass man einen robusteren Change Management-Ansatz benötigt als den Product Backlog von Scrum. Zu den Workitems gehören Anforderungen, Mängel und andere nicht funktionalitätsorientierte Arbeiten wie Schulungen, Ferien und die Unterstützung anderer Teams. Alle diese Arbeiten müssen irgendwie priorisiert werden, nicht nur die Umsetzung der Anforderungen.

Es enthält **explizite Meilensteine**. Am Ende des Lebenszyklusdiagramms finden sich Hinweise auf vorgeschlagene leichte Meilensteine, die von den Liefertteams angestrebt werden sollten. Solche Meilensteine sind ein wichtiger Aspekt agiler Governance.

DAD findet dann Anwendung, wenn mit verschiedenen Kundengruppen gearbeitet wird und

agile Vorgehensmodelle in bestehende Unternehmensstrukturen eingeführt werden.

DAD sollte in folgenden Fällen angewendet werden:

- Die Arbeit kann frühzeitig im Projekt identifiziert, priorisiert und geschätzt werden.
- Eine gute Wahl für neue agile Teams.
- Das Team ist mit Scrum und XP vertraut.
- Das Team arbeitet typischerweise an einem Projekt.

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	Was ist Disciplined Agile Delivery? . . . . .	4
1.1.1	Phasen in DAD . . . . .	5
1.1.2	DAD als Scrum-basiertes Vorgehensmodell . . . . .	6
1.2	Anwendungsgebiete der beiden Vorgehensmodelle . . . . .	7
<b>2</b>	<b>Gegenüberstellung</b>	<b>8</b>
2.1	Planung . . . . .	8
2.1.1	Release-Planung . . . . .	8
2.1.2	Priorisierung . . . . .	9
2.1.3	Planungsumfang . . . . .	10
2.1.4	Planungsaufwand . . . . .	10
2.1.5	Risikomanagement . . . . .	11
2.2	Zusammenarbeit . . . . .	12
2.2.1	Koordination im Team . . . . .	12
2.2.2	Rolle und Aufgaben des Kunden . . . . .	13
2.2.3	Kommunikation im Team . . . . .	15
2.2.4	Know-How-Transfer im Team . . . . .	16
2.2.5	Organisation in grösseren Teams . . . . .	16
2.3	Zuständigkeit . . . . .	17
2.3.1	Rollen im Team . . . . .	18
2.3.2	Verantwortlichkeiten . . . . .	21
2.3.3	Zielerfüllung . . . . .	22
2.3.4	Rollenkonflikte . . . . .	23
<b>3</b>	<b>Empfehlung</b>	<b>25</b>
<b>4</b>	<b>Interviewfragen</b>	<b>27</b>
<b>5</b>	<b>Anhang</b>	<b>28</b>

# 1. Einführung

Diese Arbeit zeigt auf wie DAD auf Scrum angewendet wird und welchen Mehrwert dadurch gewonnen wird. Wir fokussieren uns explizit in dieser Arbeit nur auf Scrum, da ein Vergleich mit allen agilen Methoden nicht umsetzbar gewesen ist. Zudem ist durch den Auftrag gegeben, dass Scrum vom bestehenden Team bereits angewendet wird. Wir empfehlen jedoch als zusätzliche Quelle das Buch «Choose your WoW! A Disciplined Agile Delivery Handbook for Optimizing Your Way of Working» [2].

## 1.1 Was ist Disciplined Agile Delivery?

Disciplined Agile Delivery (DAD) ist ein Hybrid-Prozess beziehungsweise ein Framework, welches agile Vorgehensmodelle, wie beispielsweise Scrum, integriert. Die Erfinder von DAD (Scott Ambler und Mark Lines) sehen agile Prozesse als nicht voll umfänglich. Scrum beantwortet viele Fragen nicht, welche sich stellen, wenn das Modell in einem komplexeren Umfeld angewendet wird. DAD bietet zusätzliche Fragestellungen und Methoden um gerade komplexe Strukturen auf einen agilen Pfad zu bringen. DAD ist somit eine Ergänzung zu den agilen Vorgehensweisen wie Scrum, Extreme Programming, Kanban, Lean, etc. DAD ermöglicht bestehende agile Prozesse auf komplexere Unternehmenstrukturen anzuwenden.

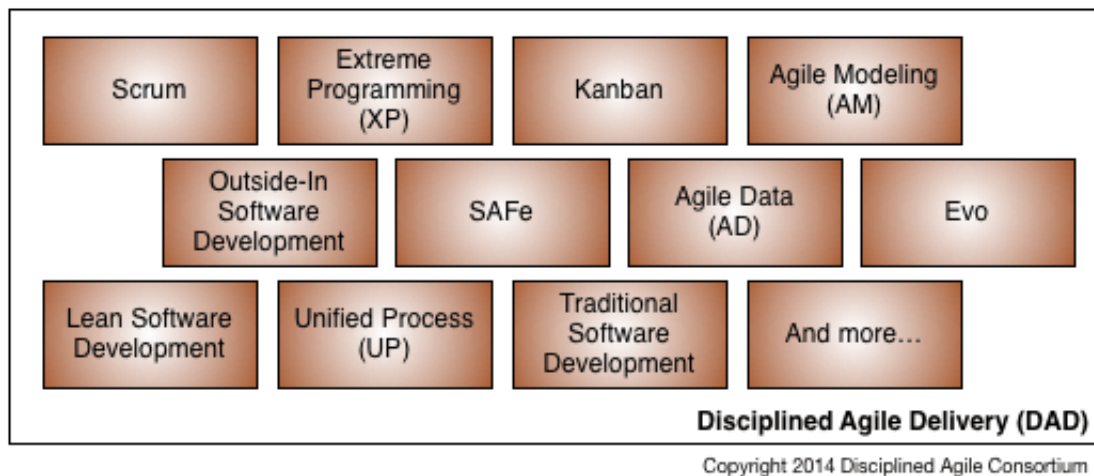


Abbildung 1.1: DAD als hybrides Vorgehensmodell [3]

### 1.1.1 Phasen in DAD

Allgemein kann gesagt werden, dass DAD aus drei Phasen besteht: Inception, Construction und Transition (Abbildung 1.2).

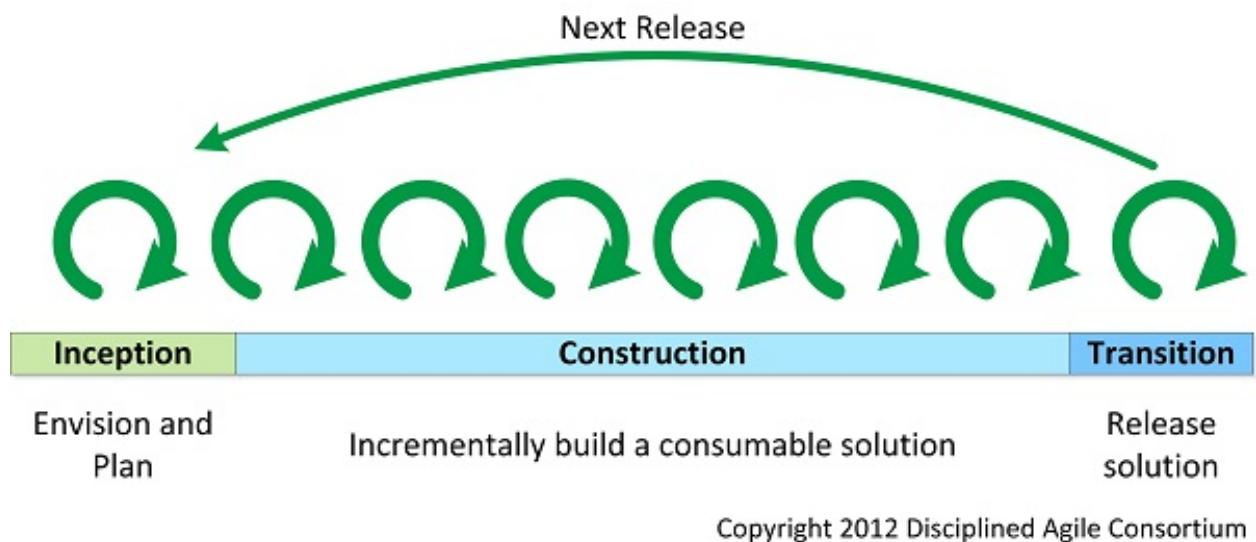


Abbildung 1.2: Highlevel Lifecycle von DAD [4]

In der **Inception** Phase wird die Planung und Analyse des Projekts und dessen Ressourcen gemacht. Dies umfasst folgende Schritte:

- Initiales Team bilden
- Projekt Vision identifizieren

- Mit Stakeholder auf die Projekt-Vision einigen
- Auf Unternehmensstrategie abstimmen
- Technische Strategie, initiale Anforderungen und initiale Release-Planung festlegen
- Arbeitsumfeld einrichten
- Finanzierung sichern
- Risiken identifizieren

Die **Construction** Phase beinhaltet die eigentliche Entwicklung und das Testen, wo das entsprechende Vorgehensmodell eingesetzt wird. Sie umfasst folgende Punkte:

- Eine verwendbare Lösung liefern
- Ändernde Bedürfnisse der Stakeholder adressieren
- Näher an das einsetzbare Produkt herankommen
- Qualität verbessern oder höhere Qualität erarbeiten
- Architektur frühzeitig beweisen
- Arbeitsumfeld einrichten

Die **Transition** Phase betrifft Zieleinhaltung und Lieferung. Folgende Punkte gehören zu dieser Phase:

- Einsatzfähigkeit der Lösung sicherstellen
- Empfangsbereitschaft der Stakeholder sicherstellen
- Lösung in produktive Umgebung liefern

Auffallend ist hier, dass die Phasen identisch mit jenen von RUP (Rational Unified Process) sind [5]. DAD beinhaltet als nebst den Fundamenten von Scrum und XP auch jene von RUP.

### 1.1.2 DAD als Scrum-basiertes Vorgehensmodell

DAD als agiles Vorgehensmodell bedeutet eine erweiterte Scrum Vorgehensweise. Scrum wird dort erweitert wo es unzureichend definiert ist. In Abbildung 1.3 ist der komplette Lifecycle des Scrum-basierten DAD dargestellt. Diese Abbildung scheint auf den ersten Blick sehr komplex. Bei genauerer Betrachtung ist aber ersichtlich, dass der Lifecycle viele Gemeinsamkeiten mit jenem von Scrum hat.

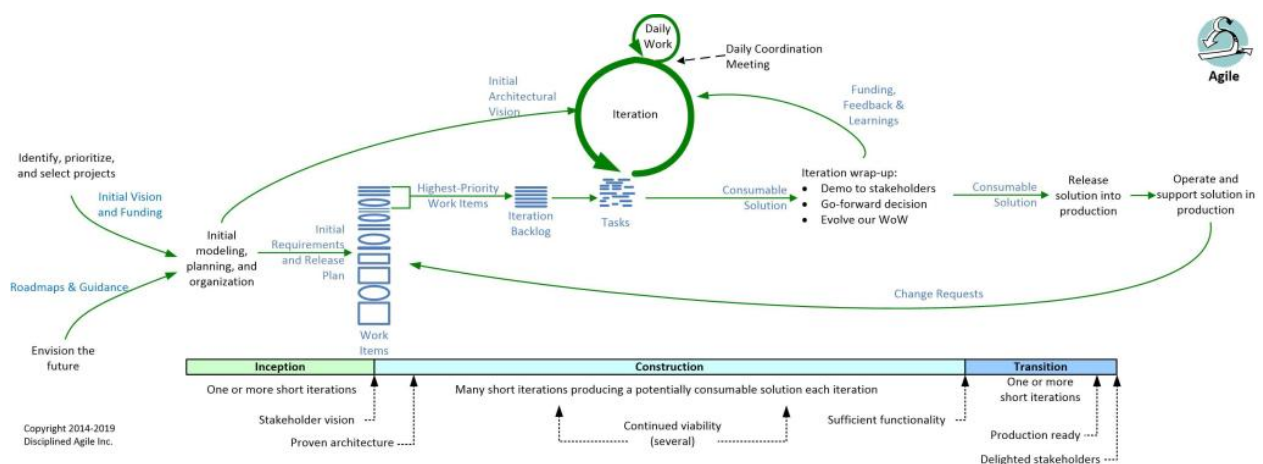


Abbildung 1.3: Scrum-basiertes DAD Vorgehensmodells [4]

Dieses Vorgehensmodell bietet folgende interessante Aspekte:

- Es ist iterationsbasiert.
- Es verwendet keine Scrum-Terminologie.
- Es zeigt Inputs ausserhalb des Vorgehensmodelles an.
- Es gibt eine Workitem-Liste, kein Product Backlog.
- Es enthält explizite Meilensteine.

## 1.2 Anwendungsgebiete der beiden Vorgehensmodelle

Scrum wird häufig im Kontext von IT, Software-Entwicklung, Design oder Marketing oder kontextähnlichen Gebieten eingesetzt. Jedoch ist wichtig, dass Scrum in allen Branchen zur Anwendung kommt.

DAD kann im selben Kontext wie Scrum genutzt werden. Jedoch ist die Anwendung nur in komplexeren Unternehmenstrukturen sinnvoll. Zudem ist DAD geeignet als Mittel zur Transition von hierarchisch geprägten Strukturen zu agilen, selbstständigen Teams.



## 2. Gegenüberstellung

### 2.1 Planung

Das Thema Planung wird unter folgenden fünf Aspekten betrachtet:

1. Release-Planung
2. Priorisierung
3. Planungssicherheit
4. Planungsaufwand
5. Nachvollziehbarkeit

#### 2.1.1 Release-Planung

Der Aspekt «Release-Planung» wurde gewählt, weil Scrum keine Aussage über die Release-Planung im weiteren Kontext aussagt. Wie wird eine abteilungsübergreifende Planung gemacht? Wie wird das Management informiert? Diese Fragen werden in diesem Kapitel beantwortet.

Scrum: [6]

Der Release-Plan ist ein höhergestellter Plan, der mehrere Sprints beinhaltet und während der Release-Planung festgelegt wird. Der Plan definiert welche Features umgesetzt werden und wann diese erfüllt sind. Er dient auch dazu den Fortschritt innerhalb des Projekts verfolgen zu können. Es können mehrere Releases während des Projekts geplant werden oder einfach ein finales Release am Ende des Projekts.

Um eine Release-Planung durchführen zu können, muss Folgendes bekannt sein:

- Ein priorisiertes Scrum-Backlog
- Die Ressourcen des Scrum-Teams
- Zielerfüllungsbedingungen

Ein Release-Plan kann Termin- oder Feature-geführt sein.

Bei Termin-geführten Projekten wird spezifiziert, welche Features bis zu einem bestimmten Termin erfüllt werden können.

Kommt es bei einem Termin-geführten Projekt zu Verzögerungen, so muss gegebenenfalls der Termin angepasst werden. Dies muss in Abstimmung mit dem Kunden gemacht werden.

Bei Feature-geführten Projekten wird spezifiziert, bis zu welchem Termin das Feature erfüllt ist. Kommt es bei Feature-geführten Projekten zu Verzögerungen, so muss zusammen mit dem Kunden besprochen werden, welche Features gegebenenfalls weggelassen werden können oder entsprechend angepasst werden müssen.

Wie der Backlog ist auch der Release-Plan bei Scrum nicht statisch. Dieser kann sich mit dem Backlog ändern oder auch nach jedem Sprint wieder diskutiert und überarbeitet werden.

DAD: [7]

In DAD wird die Release Planung initial in der Inception Phase gemacht. Der Leitfaden empfiehlt für die Releaseplanung folgende sechs Fragen zu beantworten:

- Wer wird an der Planung beteiligt sein?
- Was ist der Umfang unseres Planungsaufwands?
- Was ist unsere Gesamtstrategie, die diesen Plan vorantreibt?
- Wie detailliert sollte unser Plan sein?
- Welche Kadenzen wird das Team annehmen?
- Welchen Ansatz zur Schätzung werden wir wählen?

Damit soll sichergestellt werden, dass grundlegende Managementfragen gegenüber den Stakeholdern beantwortet sind. Zudem wird erreicht, dass eine durchführbare Strategie besteht und zwischen Stakeholder und Delivery Team ein gemeinsames Verständnis existiert.

### 2.1.2 Priorisierung

Die «Priorisierung» wurde als Aspekt gewählt, weil wichtig ist wer priorisiert und es wichtig ist das Riskio von Abhängigkeiten zwischen den Aufgaben zu reduzieren bzw. zu vermeiden.

Scrum: [8]

Das Scrum-Team priorisiert zusammen mit dem Product-Owner die Tasks/Stories aus dem Scrum-Backlog. Wichtig dabei ist, dass nicht nur priorisiert, sondern auch sortiert werden

muss. Beim Sortieren wird auch die Reihenfolge von Abläufen berücksichtigt. Die Priorisierung geht mit der Sortierung Hand in Hand.

Weiter achtet Scrum auch darauf, dass die wertvollsten Inkremente frühestmöglich umgesetzt werden. Eine Bewertung oder Posteriorisierung von Tasks kann über diverse Verfahren ermittelt werden (z.B. Kano, MoSCoW, usw.). Grundsätzlich werden immer der Nutzen und Aufwand berücksichtigt um eine Einschätzung dazu zumachen.

DAD: [9]

Die Priorisierung bei DAD verhält sich ähnlich wie die Release-Planung. Grundsätzlich gilt das Rolling-Wave-Modell. Das heisst, dass höher priorisierte Features detaillierter spezifiziert werden und tiefer priorisierte nur grob. Die Priorisierung kann jederzeit wieder angepasst werden.

### 2.1.3 Planungsumfang

Für die Planung ist ausschlaggebend, welchen Umfang für die Planung berücksichtigt wird. Das heisst, welche Arbeiten bzw. Aufwände fallen in die Planung oder nicht.

Scrum:

In Scrum betrifft der Umfang immer direkt das Produkt. Der Scope beinhaltet Features ausgedrückt z.B. als User Stories. Diese sind im Scrum-Backlog abgelegt und verwaltet.

Man kann in Scrum nicht-funktionale Punkte wie Ferien oder Schulungen auch berücksichtigen. Z.Bsp. muss die Kapazität für einen Sprint angepasst werden wenn ein Teammitglied währenddessen im Urlaub ist. Oder Eine Schulung des Betreiber-Teams kann als User-Story erfasst werden. Jedoch sind solche Anforderungen nicht implizit in Scrum angedacht.

DAD: [10]

DAD geht hier einen Schritt weiter und definiert nicht nur Features sondern sogenannte Working-Items. Bei denen werden auch nicht-funktionale Anforderungen definiert wie z.B. Schulungen, Ferien, Unterstützung anderer Teams usw.

### 2.1.4 Planungsaufwand

Im Rahmen der Planung stellt sich die Frage, welchen Aufwand die Planung braucht. Ist es ein wiederholender Prozess mit immer gleichem Aufwand oder ist es ein einmaliger initialer Prozess.

Scrum:

Der Planungsaufwand von Scrum ist relativ gering und ist im iterierenden Prozess von Scrum bereits integriert. Die Planung wird bei Scrum vor jedem Sprint im sogenannten Sprint

Planning gemacht. Dabei werden die zu erledigenden Items definiert. Schlussendlich wird der Umfang des Sprints vom ganzen Team bestätigt.

DAD:

Initial ist der Planungsaufwand bei DAD höher als in Scrum. Man muss nebst der eigentlichen Planung des Produkts auch diverse Analysen von Ist-Zuständen bezüglich Ressourcen und Zuständen innerhalb des Unternehmens machen um die Rahmenbedingungen für das Projekt zu legen. Dies bringt der Vorteil, dass in einem vielleicht eher unerfahrenen Scrum Team, auch nicht-funktionale Anforderungen berücksichtigt werden. Auch hat so Scrum in einem klassischen Unternehmen mehr Akzeptanz, da es andere Abteilungen und Prozesse berücksichtigt und mit einbezieht.

Während der Construction Phase ist der Planungsaufwand analog demjenigen von Scrum.

### 2.1.5 Risikomanagement

Bei jeder Planung stellt sich auch die Frage des Scheiterns. Dieser Aspekt zeigt wie mit den Risiken umgegangen wird.

Scrum:

Bei Scrum wird das Risikomanagement hauptsächlich durch die Kommunikation zwischen dem Kunden und Team geführt. Diese geht über den Product Owner. Dabei muss der Kunde durch seinen stetigen Einfluss mögliche Risiken ausschliessen können. Er kann dies mittels Akzeptanzkriterien beeinflussen.

Aus Team-interner Sicht ist die Definition of Done das Kontrollinstrument, um Qualität aber auch Vollständigkeit sicherzustellen.

Als weiteres Instrument dient die Review am Ende jedes Sprints. An jener werden dem Kunden die umgesetzten Items präsentiert und er kann Einfluss nehmen, sowie Missverständnisse aufdecken und klären.

DAD:

Um das Risiko von Fehlkommunikation zu verringern werden bei DAD gegenüber Scrum leichte Meilensteine eingeführt, bei denen ein Abgleich mit dem Kunden stattfindet.

Weiter sieht DAD die Planung von festen Releases vor. Damit soll regelmässig Software zur Verfügung gestellt werden, um ein Feedback des Kunden zu erhalten und frühzeitig festzustellen, ob man die Anforderungen so erfüllen kann. Dies ist gleich wie bei Scrum.

## 2.2 Zusammenarbeit

Das Thema Planung wird unter folgenden fünf Aspekten betrachtet:

1. Koordination im Team
2. Rolle und Aufgaben des Kunden
3. Kommunikation im Team
4. Know-How-Transfer im Team
5. Organisation in grösseren Teams

### 2.2.1 Koordination im Team

Die Koordination im Team wurde als Aspekt gewählt, weil es interessiert wie sich ein Team organisiert und welche Methoden/Prozesse vom Vorgehensmodell mitgegeben werden um dies zu fördern.

Scrum:

Scrum geht grundsätzlich von einer selbstorganisierten Koordination aus. Einzig der Scrum Master hat eine koordinierende Rolle, jedoch nur im Bezug auf den Prozess und Schnittstellen ausserhalb des Projekts. Als wichtigstes Instrument für Koordination in Scrum dient die Kommunikation mit den anderen Team-Mitgliedern. Dies wird in Scrum mit regelmässigen Treffen/Aussprachen realisiert. Wie zum Beispiel das Daily Stand-up oder dem «Scrum of Scrums». Beim Scrum of Scrums treffen sich Vertreter aus einzelnen Scrum-Teams um eine teamübergreifende Koordination zu haben. Weitere Routinen wie Daily Scrum oder die Reflexion dienen auch der Koordination und deren Verbesserung, auch wenn das nicht primär ihr Ziel ist. Somit liegt die Verantwortung für die Koordination seiner Aufgaben bei jedem einzelnen Teammitglied selbst.

DAD: [11]

DAD möchte, dass folgende Fragen bezüglich Koordination in einem Scrum-Team geklärt sind, um eine effektive Organisation innerhalb des Teams zu haben.

- Wie werden Informationen innerhalb des Teams ausgetauscht?
- Wer darf die vom Team erstellten Artefakte aktualisieren?
- Wie werden wir uns innerhalb des Teams koordinieren?
- Falls wir Teil eines größeren Teams sind, wie werden wir dann innerhalb dieses Teams koordinieren?
- Wie werden wir mit Enterprise-/IT-Teams wie Enterprise Architects und Data Managern zusammenarbeiten?

- Wie werden wir unsere Release-/Einsatzplanung mit dem Rest des Unternehmens koordinieren?
- Falls wir geografisch verteilte Teammitglieder haben, wie werden wir dann mit ihnen zusammenarbeiten?

Innerhalb des Teams soll also klar definiert sein, mit welchen Routinen (tägliches Treffen, Video-Konferenzen, usw.) Informationen zwischen den Teammitgliedern ausgetauscht werden und welche Tools dazu verwendet werden. Auch soll geregelt sein, wer die finalen Artefakte verwaltet, dass es hier keine Überschneidungen oder Unklarheiten gibt. Das sind Fragen, welche bei Scrum nicht implizit geklärt werden. Daher können diese Fragen gerade bei eher unerfahren Teams von Vorteil sein. Innerhalb des Teams soll also klar definiert sein, mit welchen Routinen Informationen zwischen den Teammitgliedern ausgetauscht werden und welche Tools dazu verwendet werden. Auch soll die Zuständigkeit geregelt sein wer die finalen Artefakten verwaltet. Da es sonst zu Überschneidungen oder Unklarheiten kommen kann.

Weiter muss gerade bei agilen Teams die Koordination mit anderen Teams oder Abteilungen innerhalb eines Unternehmen geregelt sein.

Was oft vernachlässigt wird, ist die Koordination mit Teammitglieder an anderen geografischen Orten. Der Informationsaustausch wird hier anspruchsvoller, da direkte verbale Kommunikation, welche die effektivste ist, nicht möglich ist.

### 2.2.2 Rolle und Aufgaben des Kunden

Die Rolle und Aufgabe des Kunden wurde als Aspekt gewählt, weil es interessiert wie der Kunde im Projekt/Prozess miteinbezogen wird.

Scrum:

Der Kunde soll in Scrum während des ganzen Projekts immer involviert sein. Folgende Möglichkeiten gibt es den Kunden zu involvieren:

- Der Kunde wird zum Initial-Meeting eingeladen.
- Das Backlog wird zusammen mit dem Kunden verwaltet.
- Der Kunde nimmt auch an Reviews teil, um Arbeiten zu besprechen und als abgeschlossen zu definieren.

Durch den konsequenten Miteinbezug des Kunden wird das Risiko vermindert, dass der Kunde nicht zufrieden ist. Dies weil er fortlaufend Einfluss nehmen kann und mit seiner Teilnahme auch frühe Schritte/Arbeiten bestätigt.

## DAD: [12]

DAD hat bezüglich dem Kunden eine andere auch radikalere Haltung. In DAD will man den Begriff «Kunde» nicht verwenden, sondern nur Stakeholder. Diese werden in folgende Gruppen unterteilt:

- Endbenutzer: Personen die das Produkt schlussendlich verwenden.
- Vorstehende: Personen die schlussendlich entscheiden, welches Produkt beschafft wird, Bezahlungen freigeben, usw.
- Partner: Unterhalter, Betreiber, Entwickler von externen Systemen, Juristen, usw.
- Interne: Personen innerhalb des Entwicklungsteams und welche technische oder geschäftliche Dienste liefern

Für ein Produkt gibt es nur Stakeholders und es gilt dessen Anforderungen genau zu ermitteln und festzulegen. Dazu werden für alle Stakeholder die Bedürfnisse gleichwertig ermittelt und miteinbezogen.

Man will also bewusst ein Produkt, das alle Stakeholder gleich berücksichtigt und nicht nur den «bezahlenden Kunden» hauptsächlich priorisiert. Denn nur so wird der Kunde auch ein nachhaltiges und erfolgreiches Produkt erhalten können.

Wichtig ist, dass das Projekt im «wir»-Kontext betrachtet wird und nicht im «ihr»-Kontext. Es gibt nicht den Kunden und das Entwicklungsteam, sondern das Projekt betrifft alle gleich.

### 2.2.3 Kommunikation im Team

Die Kommunikation wurde als Aspekt gewählt, da es wichtig ist wie kommuniziert wird um eine effiziente Zusammenarbeit zu gewährleisten, gerade bei geographisch getrennten Teams.

Scrum:

Scrum schreibt nicht vor wie, also die Form, die Kommunikation innerhalb des Teams stattfinden soll. Scrum sagt zur Kommunikation lediglich, dass sie stattfinden muss und sie essenziell für den Projekterfolg ist. Scrum bezieht sich dabei auch auf das Agile Manifest:

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. [13]

DAD: [14]

In DAD wird die Kommunikationsart dem Informationsgehalt gegenübergestellt. Grundsätzlich ist die direkte Kommunikation, Face-to-Face, am effizientesten. Jedoch müssen auch andere Faktoren und Anforderungen dabei berücksichtigt werden. Sind beispielsweise verschiedene Standorte in einem Team vertreten, kann eine direkte Kommunikation aufwändiger und gegebenenfalls teurer werden. So kann zusätzliche Infrastruktur benötigt werden. Oder die Kommunikation kann aufgrund unterschiedlicher Zeitzone erschwert sein. DAD bietet hier aber einige Ideen für die effiziente Kommunikation wie in Abbildung 2.1 gezeigt.

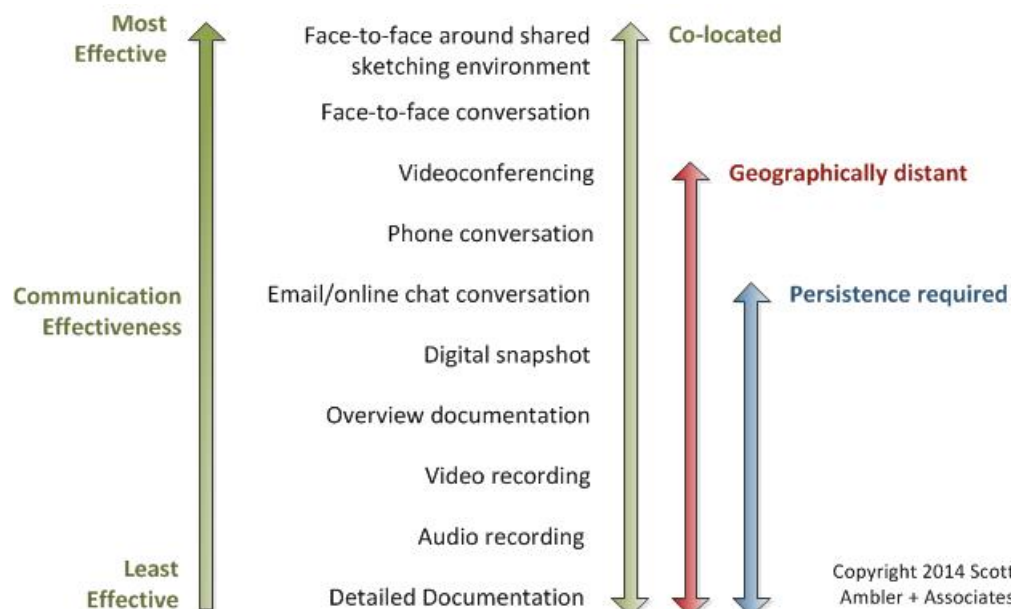


Abbildung 2.1: DAD Kommunikationstechniken



## 2.2.4 Know-How-Transfer im Team

Know-How-Transfer im Team wurde als Aspekt gewählt, weil DAD dazu spannende Ideen bringt wie Kompetenzen und Wissen auf verschiedene Arten ausgetauscht werden kann.

Scrum:

Scrum macht direkt keine Vorschläge zum Austausch von Know-How. In den Reviews kann ein gewisser Austausch über Know-How stattfinden, jedoch ist dort der Fokus primär auf die Zusammenarbeit gerichtet.

DAD: [15]

DAD macht einige Vorschläge wie Teammitglieder sich gegenseitig Know-How oder Kompetenzen vermitteln können.

Folgende Methoden werden vorgeschlagen:

- Training: sich etwas beibringen
- Open Space: gemeinsam über Fragestellungen diskutieren
- Non-Solo work: bewusst Arbeiten nicht alleine machen
- Mentoring: sich von jemandem etwas erklären lassen
- Hackathons: z.B. gemeinsam in einem Tag eine Applikation für andere Anwendung entwickeln oder gemeinsam einen Task umsetzen (Mob-Coding)
- Communities of Practice: Treffen von Personen aus verschiedenen Abteilungen, welche ähnliche Problemstellungen im Berufsalltag haben

Mit solchen Methoden kann spielerisch und immer wieder in einzelnen Blöcken Know-How und Kompetenzen zwischen den einzelnen Mitarbeitern ausgetauscht werden. Weiter können solche Anlässe auch einen Team-bildenden Effekt haben.

## 2.2.5 Organisation in grösseren Teams

Die Organisation in mittleren bis grossen Teams wurde als Aspekt gewählt, weil Scrum hier eine Limitierung von max. 7-9 Leuten hat. Es interessiert welchen Ansatz DAD bringt um agile Vorgehensmodelle wie Scrum auch in grösseren Teams anzuwenden.

Scrum: [16]

Scrum geht von einer maximalen Grösse eines Teams von 7-9 Mitgliedern aus. Danach wird das Projekt mit all seinen Prozessen zu lange und ineffizient (z.Bsp. Daily Scrum nicht mehr in 15min möglich).

Es wird dann eine Aufteilung des Teams empfohlen und das Projekt entsprechend zu modularisieren. Um einen Austausch zwischen diesen Teams zu haben soll übergeordnetes Scrums of Scrums gehalten werden, wo nur die einzelnen Product Owner und Scrum Master teilnehmen um ihre Teams zu vertreten.

DAD: [17]

DAD unterscheidet zwischen kleinen (2-15 Personen )mittleren(15-35 Personen) und grossen (35+ Personen) Teams.

Kleine Teams werden wie in Kapitel 2.3.1 beschreiben geführt. Also grundsätzlich Scrum mit zusätzlichen Rollen.

Mittlere Teams werden ähnlich dem Scrum of Scrums geführt.

Grosse Teams werden in DAD zusätzlich definiert. Hierzu wird zu den einzelnen Subteams ein Leadership Team eingeführt. Dieses enthält folgende zusätzlichen Rollen:

- Program Manager: hat die Koordination über das ganze Projekt und das Leadership Team
- Product Coordination: Koordiniert alle Team Leader aus den Sub-Teams
- Product Ownership: Koordiniert alle Product Owner aus den Sub-Teams
- Architecture Ownership: Koordiniert alle Achitecture Owner aus den Sub-Teams

Das Leadership Team soll die einzelnen Sub-Team und die Rollen-Inhaber in ihren Tätigkeiten unterstützen und die Koordination zwischen den Teams sicherstellen und unterstützen.

## 2.3 Zuständigkeit

Das Thema Zuständigkeiten wird unter folgenden fünf Aspekten betrachtet:

1. Zuständigkeiten im Team
2. Rollen im Team
3. Verantwortlichkeiten
4. Zielerfüllung
5. Rollenkonflikte

Auf den Aspekt «Zuständigkeiten im Team» wird nicht explizit eingegangen. Die Zuständigkeiten im Team sind über die weiteren viere Aspekte verteilt zu betrachten.

### 2.3.1 Rollen im Team

Der Aspekt «Rollen im Team» wurde gewählt, dass für den Prozess ausschlaggebend ist, wer am Prozess beteiligt ist. Welche Rollen gibt es im jeweiligen Prozess und welche Funktion wird der jeweiligen Rolle zugewiesen?

Scrum: [18]

In Scrum gibt es drei Rollen, siehe Abbildung 2.2. Den Scrum Master, den Product Owner und die Team Member.

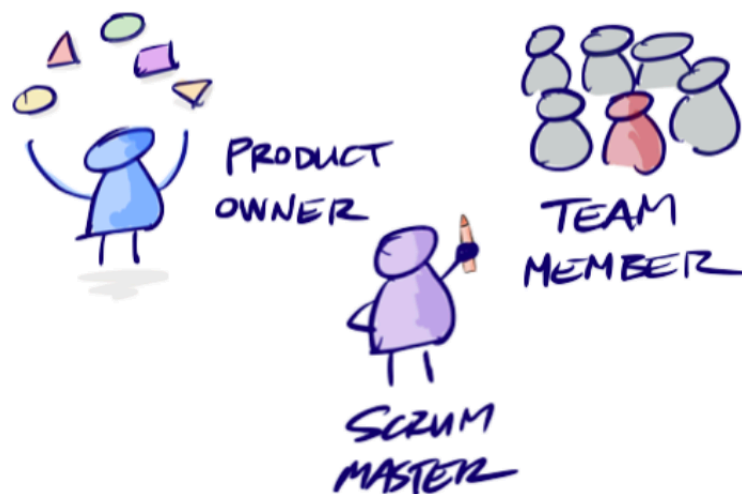


Abbildung 2.2: Scrum Rollen [18]

Der **Product Owner** ist die Ansprechperson für Kunden und Stakeholder. Er nimmt die Anforderungen entgegen, priorisiert diese und gibt diese an das Team weiter. Der Product Owner kann aber auch direkt der Kunde sein. In beiden Fällen vertritt er aber die fachliche Sicht, beurteilt die Qualität, Usability und Performance.

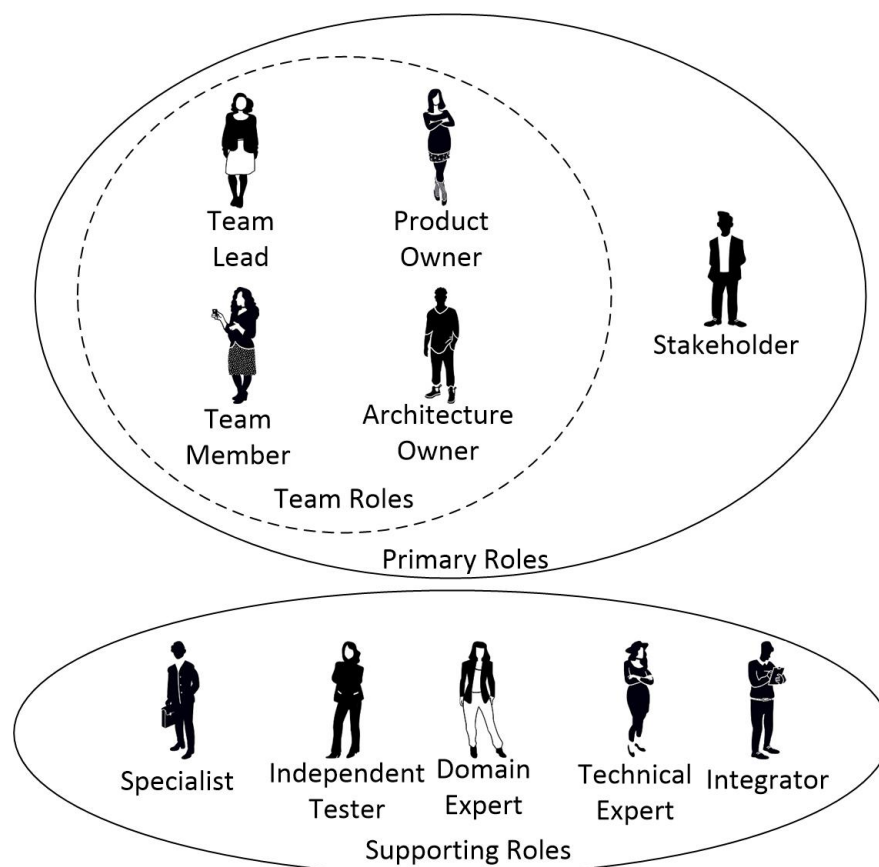
Der **Scrum Master** dient dem Scrum Team um die gewünschte Performance zu erreichen. Er ist verantwortlich, dass der Prozess zielgerichtet ausgeführt werden kann. Er nimmt sich Problemen, sogenannten Impediments, an und löst diese - mit oder ohne Team. Der Scrum Master fördert den Fortschritt des Teams und lenkt dieses, falls der Fortschritt nicht wie geplant vorankommt. Zudem organisiert er die Scrum Zeremonien - Retrospektive, Review-Meeting, Daily Standups - und sorgt für den Informationsfluss zwischen Team und Product Owner. Er ist sozusagen die gute Seele des Teams.

Das **Scrum Team**, oder kurz Team, ist das zentrale Element im Scrum Prozess. Es setzt die Anforderungen um. In Scrum besteht ein Team aus fünf bis zehn Personen. Grössere Teams sollten in mehrere unabhängige Teams aufgetrennt werden. Sehr wichtig bei Scrum ist, dass das Team selbstorganisiert ist. Es bestimmt schlussendlich selbst welche Anforderungen es

im Sprint umsetzen will. Dabei beachtet es, dass es am Ende jedes Sprints ein «Increment of Potentially Shippable Functionality» liefert. In einem Scrum Team sind immer die Disziplinen vorhanden, welche es für das Erreichen des Ziels benötigt.

## DAD: [19]

Im Gegensatz zu Scrum werden in DAD zusätzliche Rollen definiert, siehe Abbildung 2.3. In DAD werden einzelne Zuständigkeiten explizit aufgeführt, welche in Scrum im Team vereint werden. DAD verwendet gewisse Rollen die auch Scrum verwendet (Team Member und Product Owner), aber unterscheidet hierbei primäre und unterstützende Rollen. Zu den primären Rollen zählen Team Lead, Product Owner, Team Member, Architecture Owner und Stakeholder. Zu den unterstützenden Rollen zählen Spezialisten, Independent Tester, Domänen-Experten, Technische Experten und Integratoren.



Copyright 2012-2019 Disciplined Agile, Inc.

Abbildung 2.3: DAD Rollen [19]

DAD wurde somit stark auf die Bedürfnisse von Unternehmen weiterentwickelt um Anforderungen aus den bestehenden Firmenstrukturen, mit Prozessen, Programmen und Rollen, anzuwenden. DAD unterstützt mit den sehr spezifischen Rollen insbesondere hierarchische Unternehmensstrukturen.

In diesem Punkt unterscheiden sich die zwei Methoden wesentlich. Warum hat Scrum drei Rollen, DAD hingegen zehn? Scrum konzentriert sich hauptsächlich auf Führungs- und Change-Management-Aspekte während der Konstruktion und hat daher Rollen, die dies

widerspiegeln. DAD hingegen konzentriert sich explizit auf den gesamten Delivery Lifecycle und alle Aspekte der Lösungsbereitstellung, einschließlich der technischen Aspekte, die Scrum auslöst. Mit einem grösseren Umfang kommen also mehr Rollen hinzu. Weil DAD beispielsweise Fragen der agilen Architektur umfasst, beinhaltet es auch eine Rolle des Architecture Owners. Scrum adressiert keine Architektur und beinhaltet auch keine solche Rolle.

### 2.3.2 Verantwortlichkeiten

In diesem Aspekt wird beschrieben wie die Verantwortlichkeiten auf die Rollen verteilt sind und welche Verantwortlichkeiten im Prozess wahrgenommen werden müssen.

Scrum: [20]

In Scrum hat jede Rolle ihre Verantwortlichkeit im Prozess. Der Scrum Master hat in erster Linie die Verantwortung zur Einhaltung des Prozesses und der Scrum Spielregeln. Wie bereits im Kapitel 2.3.1 beschrieben, hat er die Verantwortung zur Beseitigung von Hindernissen.

Der Product Owner hat die Verantwortung die Vision und Wünsche des Kunden an das Team weiterzugeben. Er ist die Stimme der Stakeholder und somit verantwortlich die Bedürfnisse des Kunden zu identifizieren.

Die grösste Verantwortung wird jedoch dem Team zugeteilt. Es legt den Umfang fest, welchen es umsetzt - das sogenannte Sprint Goal. Es verpflichtet sich dazu das Sprint Goal umzusetzen. Bei Problemen oder Störungen muss das Team zeitnah mit dem Kunden bzw. mit dem Product Owner in Kontakt treten, um gegebenenfalls das Sprint Goal anzupassen. Da das Team selbstorganisiert ist, liegt es in seiner Verantwortung, wie das Ziel erreicht wird.

DAD: [21]

Da es in DAD mehr Rollen als in Scrum gibt, sind auch die Verantwortlichkeiten expliziter. Wir gehen hier nur auf die Verantwortlichkeiten der primären Rollen ein.

Der Team Lead hat in DAD eine ähnliche Funktion und Verantwortlichkeiten wie der Scrum Master. Er dient dem Team als Coach. Er schaut, dass das Team performant arbeiten kann und fördert das Team in der Verbesserung ihrer Arbeitsweise.

Der Product Owner in DAD unterscheidet sich kaum vom PO in Scrum. Auch in DAD ist er die Stimme des Kunden. Er priorisiert die Aufgaben und versucht die Fragen des Teams zu beantworten oder leitet diese an den Kunden weiter.

Der Architecture Owner ist verantwortlich für die Umsetzung der architektonischen Entscheidung. Er trifft die Architektur Entscheidung, erstellt das Design der Gesamtlösung und übernimmt die Weiterentwicklung davon. Oft ist der Architecture Owner auch der Lead Entwickler. Manchmal wird er auch Softwarearchitekt oder Solutionarchitekt genannt. Der

AO ist keine hierarchische Rolle. Er arbeitet mit dem Team und trifft die Entscheidungen in Absprache mit ihm.

Wie in Scrum hat auch das Team in DAD die grösste Verantwortung. Darunter zählen:

- Eine Lösung zu finden, die den Bedürfnissen der Stakeholder entspricht.
- Optimierte Nutzung der Ressourcen
- Bereitschaft zu einer umfassenden Zusammenarbeit innerhalb Ihres Teams, auch ausserhalb Ihres Fachgebiets.
- Informationsaustausch über alle Arbeiten, die im Gange sind.
- Andere in Ihren Fähigkeiten und Erfahrungen zu coachen.
- Erweiterung Ihrer Kenntnisse und Fähigkeiten ausserhalb Ihres Fachgebiets.
- Teilnahme an den Meetings.
- Proaktiv nach Wegen zur Verbesserung der Teamleistung suchen
- Vermeiden, dass Arbeiten außerhalb der aktuellen Iteration ohne Zustimmung des Teams angenommen werden.
- Arbeit so früh wie möglich validieren und mit anderen zusammenzuarbeiten

Diese Punkte sind ziemlich Deckungsgleich wie jene die im agilen Manifest geschildert werden.

### 2.3.3 Zielerfüllung

In jedem Projekt ist die Zielerreichung das Wesentlichste. Darum wird in diesem Aspekt erläutert wie in beiden Modellen die Zielerreichung bzw. Zielerfüllung sichergestellt wird.

Scrum: [22]

Jedes Artefakt, welches in einer Iteration (Sprint) umgesetzt wird, wird durch Akzeptanzkriterien beschrieben. Diese werden zusammen mit oder durch den Kunden definiert. Anhand dieser Akzeptanzkriterien wird ein Artefakt gemessen. Die Akzeptanzkriterien dienen dem Team, nebst einer genauen Beschreibung des Artefakts, zur Erreichung des Sprint Ziels.

In Scrum wird nach jedem Sprint die sogenannte Review gemacht. Hier wird dem Kunden das Produkt bzw. die umgesetzten Artefakte präsentiert. Hier hat der Kunde die Möglichkeit lenkend einzugreifen. Da in Scrum in kurzen Iterationen gearbeitet wird, kann der Kunde schnell und zeitnah Einfluss auf Missverständnisse nehmen.

Das Team definiert zudem die Definition of Done. Die DoD definiert, wann ein Task abgeschlossen ist. Diese Kriterien müssen nicht zwingend deckend zu den Akzeptanzkriterien sein. Oft hat die DoD umfassendere Punkte wie bspw. Tests, Dokumentation, Review etc.

## DAD: [23]

Grundsätzlich hat DAD dieselben Mechanismen zur Sicherstellung der Zielerreichung wie Scrum. Nach jeder Iteration wird, wie in Scrum auch, die Iteration zusammengefasst. Den Stakeholdern wird eine Demo der umgesetzten Tasks gezeigt und es wird entschieden ob man mit dem Vorgehen weitermacht. Zum Abschluss wird eine Analyse über den Way of Work gemacht und die nötigen Verbesserungen werden angebracht.

In DAD wird zudem eine gemeinsame Vision mit den Stakeholdern erstellt. Dies ist ein weiteres Mittel zu Sicherstellung der Zielerreichung.

In DAD besteht jede Phase, Inception, Construction, Transition, aus mehreren Goals, dargestellt in Abbildung 2.4. Ein Goal kann einmal oder mehrmals durchlaufen werden. Ein Goal besteht aus einem oder mehreren Decision Points.

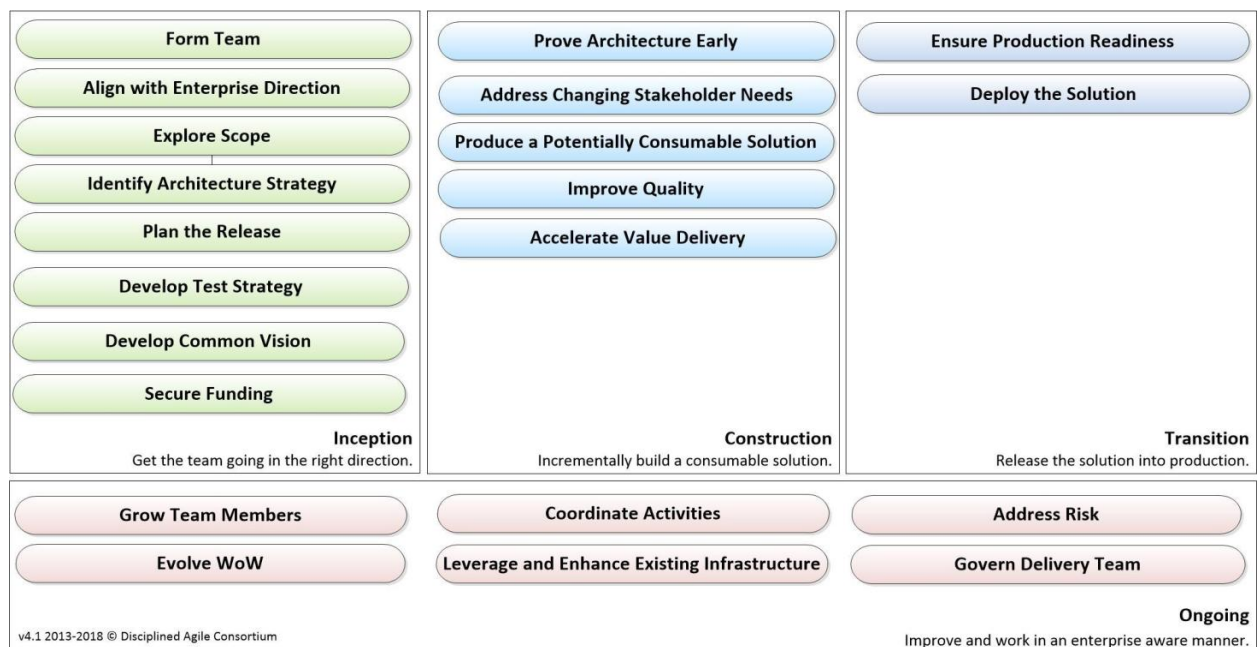


Abbildung 2.4: Lifecycle Goals [23]

Die Decision Points und Goals dienen der Sicherstellung des Prozesses und somit auch der Erreichung der Ziele.

### 2.3.4 Rollenkonflikte

Wo es Rollen gibt, gibt es auch Potential für Konflikte. Darum wird unter dem Aspekt «Rollenkonflikte» ausgeführt, wo es in den Modellen Konflikte geben kann [24]

In Scrum wie auch in DAD sind die Teams selbstorganisiert, was impliziert, dass Hierarchien in Frage gestellt werden. Gerade in Scrum werden explizite Disziplinen nicht gesucht. Scrum fordert, dass alle Mitglieder eines Teams möglichst jede Aufgabe im Sprint abarbeiten



können. Ein exklusiver Tester, Programmierer oder Architekt passt somit nicht optimal in ein Scrum Team. Ein gutes Team kann dies aber berücksichtigen und die Nachteile kompensieren. Dies sollte aber wenn möglich in erfahrenen Teams gemacht werden.

DAD bietet hier Unterstützung indem zusätzliche Rollen explizit adressiert werden. So ist nicht nur der Architecture Owner explizit aufgeführt, sondern DAD weist auch unterstützende, sogenannte Supporting Roles, auf. Diese können je nach Bedürfnis des Teams und des Projekts verwendet werden. So kann bei Bedarf ein Tester oder ein Domänen Experte hinzugezogen werden. DAD versucht hier die Rollenkonflikte zu mindern. Jedoch ist darauf hinzuweisen, dass auch DAD Hierarchien in Frage stellt und somit selbstorganisierte Teams fördert.

### 3. Empfehlung

Grundsätzlich wäre der Ansatz mit DAD sicher spannend. Ihr Team kann nach wie vor mit Scrum arbeiten und Sie haben die Möglichkeit mit den erwähnten Instrumenten wie leichten Meilensteinen oder der Release Planung Einfluss auf die agile Entwicklung zu nehmen. Somit können Sie der Entwicklung den nötigen Freiraum geben, aber trotzdem eine gezielte Steuerung vornehmen.

Jedoch wird das Einführen dieses Vorgehensmodell vergleichsweise hohe Aufwände verursachen. Vor allem muss für das saubere Anwenden von DAD viel Analyse von bestehenden Prozessen und Zuständen innerhalb Ihres Unternehmens betrieben werden.

Die Gegenüberstellung von Scrum und DAD hat gezeigt, dass mit beiden Modellen Vor- aber auch Nachteile entstehen. Aus unserer Sicht überwiegen für Ihre Unternehmung die Vorteile von DAD. Diese sollen im Folgenden nochmals Zusammenfassend aufgezeigt werden:

Im Bereich der Releaseplanung bietet DAD den Vorteil, dass die Release Planung bereits frühzeitig im Projekt definiert wird. Gegenüber Scrum werden hier viel tiefergehende Fragen geklärt um das Projekt im Einklang mit der unternehmerischen Strategie zu bringen. Dabei wird früh mit den beteiligten Stakeholdern ein gemeinsames Verständnis gefunden, nach welchem das weitere Vorgehen definiert ist. Scrum konzentriert sich hier mehr auf das Projekt und klammert unternehmerische Fragen grösstenteils aus.

Die Priorisierung von Scrum und DAD verhält sich sehr ähnlich. Bei beiden wird iterativ eine Priorisierung der Inkremente vorgenommen, wobei die wichtigsten Features jeweils den Vortritt erhalten. DaD geht zusätzlich nach dem Rolling-Wave-Modell vor und spezifiziert höher priorisierte Features detaillierter als tiefer priorisierte. Wird dieser Vorgang angewendet, so geschieht im Projekt Team dadurch eine Fokussierung aufs Wesentliche. Die wichtigsten Requirements werden genauer besprochen, während die Details von weniger wichtigen Anforderungen zeitsparend weggelassen werden können.

Der Planungsaufwand bei DAD ist initial sicher höher, dafür bietet der Planungsumfang grosse Vorteile. So werden z.B. auch nicht funktionale Anforderungen definiert, welche über das eigentliche Projekt hinaus gehen. Womit DAD im Vergleich zu Scrum besser mit bestehenden Unternehmensstrukturen und spezifischen Projektanforderungen der Unternehmen vereinbar ist.

Scrum definiert für die Zusammenarbeit verschiedene Rollen innerhalb des Projekt Teams. Das Modell klärt aber keine oder nur wenige Fragen bezüglich der Koordination mit der Unternehmung und den darin bereits vorhandenen Rollen. Hier bietet DAD einen grossen

Vorteil, da es genau auf diese Fragen eingeht. Somit können mit DAD weiterführende Fragen zur Integration des Projekt Teams in die Unternehmung geklärt werden und das Modell optimal im bestehenden Unternehmen integriert werden. So werden im DAD Modell auch einzelne Rollen, die im Scrum in den Aufgabenbereich des Teams fallen, explizit aufgeführt. Diese sind spezifisch auf die Bedürfnisse von bestehenden Firmenstrukturen ausgelegt. Insbesondere ist DAD auf hierarchische Unternehmensstrukturen ausgelegt welche durch die definierten Rollen im Modell übernommen werden können.

Gegenüber dem Kunden gibt es grosse Unterschiede zwischen Scrum und DAD. Scrum priorisiert den zahlenden Kunden, während DAD versucht alle Stakeholder gleichwertig mit einzubeziehen. Daraus ergibt sich mit DAD umfassendere Rückmeldungen zu den einzelnen Releases. Das führt zwar zu einem komplexeren Gesamtbild, dafür können Stakeholder Bedürfnisse früh herauskristallisiert und entsprechend früh in den Entwicklungsprozess miteinbezogen werden. Wird dies dem Kunden gegenüber entsprechend kommuniziert, können dadurch grosse Vorteile, sowie Kosteneinsparungen im Projekt erreicht werden.

Zusammenfassend kann festgehalten werden, dass Scrum und DAD sich zwar sehr ähnlich sind, DAD aber einige wichtige Vorteile für Ihr Unternehmen bietet. Neben den vielfältigeren Rollen, welche so ausgelegt sind, dass sie in die bestehenden Unternehmensstrukturen eingebettet werden können, werden mit DAD auch Unternehmensstrategische Ziele berücksichtigt. Dies sind wichtige Fragen, die für die beiden bestehenden Unternehmen geklärt werden müssen und genau dafür ist DAD ausgelegt. Somit kann ein Team nach wie vor mit Scrum arbeiten, da DAD eine Erweiterung von Scrum darstellt, und Sie haben die Möglichkeit mit den erwähnten Instrumenten wie leichten Meilensteinen und Releases Planung gezielt Einfluss auf die Entwicklung zu nehmen.

Ein weiterer Vorteil für die Einführung von DAD ist die Tatsache, dass sich beide Unternehmen auf neue Strukturen einlassen müssen. Was im gegebenen Fall sicher von Vorteil ist. Zwar ist der Planungs- und Transitionsaufwand höher als bei der Übernahme eines bereits verwendeten Modells, doch die gemeinsame Bewältigung der Aufgabe wird den Zusammenhalt der Teams stärken und ein gemeinsames Verständnis zwischen den beiden Unternehmen schaffen. Aus den genannten Gründen empfehlen wir Ihnen unter Berücksichtigung Ihrer bestehenden Strukturen das Modell DAD einzuführen.

## 4. Interviewfragen

- Welche Prozessmodelle kennen Sie bereits?
- Welches der Ihnen bekannten Vorgehensmodellen ist ihr persönlicher Favorit und weshalb?
- Welchem Prozessmodelltyp (linear, agil) ordnen Sie das bearbeitete Vorgehensmodell zu und warum?
- Was sind Vorteile von linearen bzw. agilen Prozessmodellen?
- Welches Modell verwenden Sie in Ihrem Betrieb?
- Was würden Sie als den grössten Vorteil des bearbeiteten Prozessmodells bezeichnen?
- Was würden Sie als den grössten Nachteil des bearbeiteten Prozessmodells bezeichnen?
  - Welche Rollen gibt es in diesem Prozessmodell?
  - Welche Rolle nehmen Sie ein oder würden Sie gerne einnehmen?
  - Wie sind Ihre Erfahrungen damit?
- Wenn Sie ein beliebiges Prozessmodell ausprobieren könnten, welches würden Sie wählen?

## 5. Anhang

Die Abbildung 5.1 zeigt den Lifecycle von Scrum.

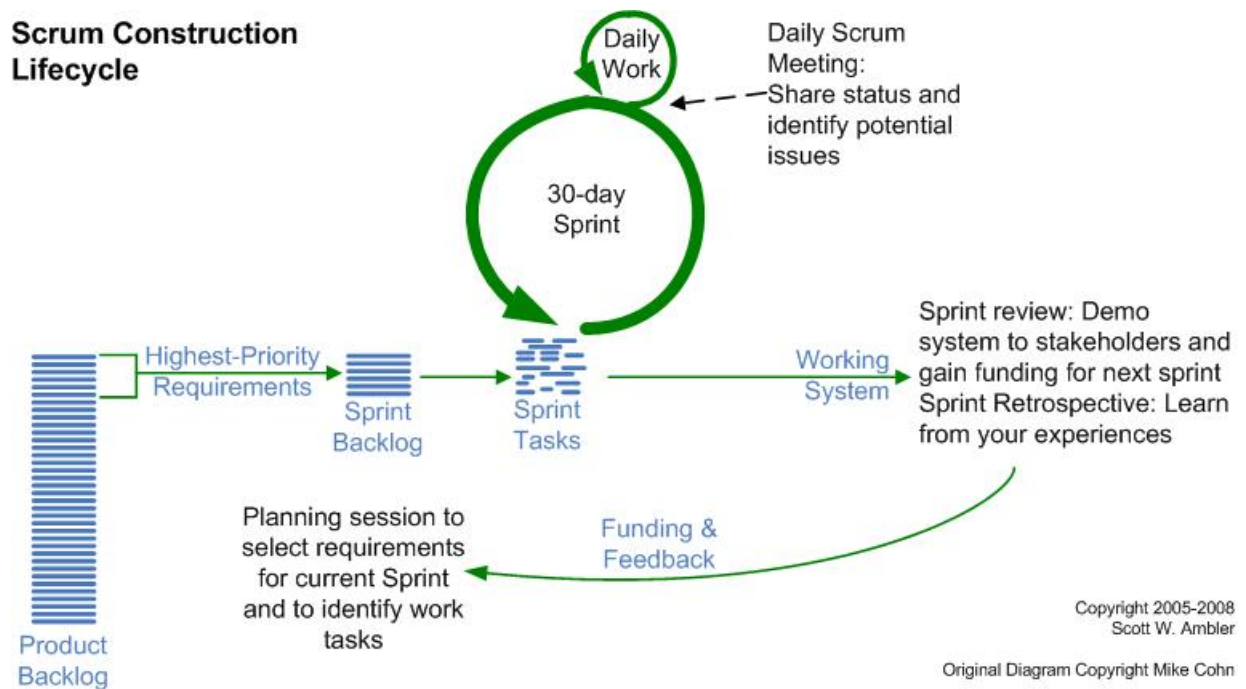


Abbildung 5.1: Der Lifecycle von Scrum [25]

# Abbildungsverzeichnis

1.1	DAD als hybrides Vorgehensmodell [3]	5
1.2	Highlevel Lifecycle von DAD [4]	5
1.3	Scrum-basiertes DAD Vorgehensmodells [4]	7
2.1	DAD Kommunikationstechniken	15
2.2	Scrum Rollen [18]	18
2.3	DAD Rollen [19]	20
2.4	Lifecycle Goals [23]	23
5.1	Der Lifecycle von Scrum [25]	28

# Literaturverzeichnis

- [1] Scott Ambler. Disciplined agile terminology. <https://disciplinedagiledelivery.com/disciplined-agile-terminology/>.
- [2] Scott Ambler Mark Lines. Book: Choose your wow! <https://disciplinedagiledelivery.com/dad-handbook/>.
- [3] Scott Ambler Mark Lines. A hybrid toolkit. <https://disciplinedagiledelivery.com/hybridframework/>.
- [4] Scott Ambler Mark Lines. Full delivery lifecycles. <https://disciplinedagiledelivery.com/lifecycle/>.
- [5] Wikipedia. Rational unified process. [https://de.wikipedia.org/wiki/Rational\\_Unified\\_Process](https://de.wikipedia.org/wiki/Rational_Unified_Process).
- [6] Author nicht erwähnt. Scrum release planning. [https://www.scrum-institute.org/Release\\_Planning.php](https://www.scrum-institute.org/Release_Planning.php).
- [7] Scott Ambler Mark Lines. Rolling wave release planning for agile delivery teams. <http://disciplinedagiledelivery.com/rolling-wave-release-planning>.
- [8] Author nicht erwähnt. Ordered not prioritized. <https://www.scrum.org/resources/ordered-not-prioritized>.
- [9] Scott Ambler. Agile core practice: Prioritized requirements. <http://agilemodeling.com/essays/prioritizedRequirements.htm>.
- [10] Scott Ambler. Dad lifecycle - agile (scrum based). <http://disciplinedagiledelivery.com/lifecycle/agile-lifecycle/>.
- [11] Scott Ambler. Coordinate activities. <https://disciplinedagiledelivery.com/coordinate-activities/>.
- [12] Scott Ambler. Stakeholder over customer. <https://disciplinedagiledelivery.com/stakeholders-over-customers/>.
- [13] Agile Manifesto. Principles behind the agile manifesto. <https://agilemanifesto.org/principles.html>.

- [14] Scott Ambler. Choose the best communication technique available. <https://disciplinedagiledelivery.com/choose-the-best-communication-technique-available/>.
- [15] Scott Ambler. Grow team member. <https://disciplinedagiledelivery.com/grow-team-members/>.
- [16] Vytas Butkus. Too big to scale – optimal scrum team size guide. <https://www.toptal.com/product-managers/agile/scrum-team-size>.
- [17] Scott Ambler. Large agile teams. <https://disciplinedagiledelivery.com/agility-at-scale/large-agile-teams/>.
- [18] agilevelocity. Scrum roles q&a recap and recording. <https://agilevelocity.com/webinar/agile-roles-webinar-recap/>.
- [19] Scott Ambler Mark Lines. Roles on dad teams. <https://disciplinedagiledelivery.com/roles-on-dad-teams/>.
- [20] Samuel Nicholson. Agile scrum roles and responsibilities. <https://www.knowledgehut.com/blog/agile/agile-scrum-roles-responsibilities>.
- [21] Glen Little. Team member responsibilities. <https://disciplinedagiledelivery.com/team-member-responsibilities/>.
- [22] Ralph Jocham. The definition of done vs acceptance criteria. <https://www.scrum.org/resources/definition-done-vs-acceptance-criteria>.
- [23] Scott Ambler Mark Lines. Process goals. <https://disciplinedagiledelivery.com/process-goals/>.
- [24] Author nicht erwähnt. Rollenkonflikte. <https://www.der-scrum-master.de/scrum-einfuehrung/scrum-rollenkonflikte>.
- [25] Scott Ambler. System development life cycle. <http://www.ambysoft.com/essays/agileLifecycle.html>.