

Kollaboratives Risikomanagement mit Dependency Poker: Beschleunigte Analyse von Abhängigkeiten und nachhaltige Lösungsstrategien

In größeren Software-Entwicklungsprojekten sind Risiko- und Abhängigkeitslisten Standard, um potenzielle Risiken und Abhängigkeiten von externen Ressourcen zu identifizieren. Besonders im agilen Projektmanagement spielen kurze Iterationen eine wichtige Rolle, um zeitnah Feedback von Stakeholdern und Benutzern zu erhalten. Unaufgelöste Abhängigkeiten stellen ein signifikantes Risiko dar, da sie den kritischen Pfad des Projektplans gefährden, der für einen Sprint oder die Fertigstellung eines MVPs¹ im Fokus steht.

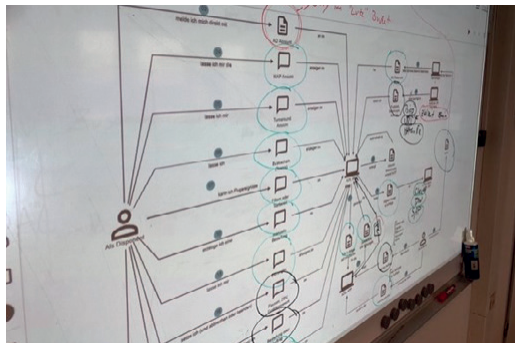


Abbildung 1: Per Domain Storytelling geplantes MVP²

Jedes cross-funktionale Team profitiert von einem erfahrenen Projektmanager, der über Kompetenzen im Risikomanagement mit dem Schwerpunkt Abhängigkeiten verfügt. Für jede identifizierte Abhängigkeit oder abgeleitete Risiko werden die Schwere der Auswirkungen und die Wahrscheinlichkeit des Eintritts bewertet, um geeignete Maßnahmen zu ergreifen. Im agilen Vorgehen in kurzen Iterationen gearbeitet um Hypothesen über die komplexe Umwelt überprüft werden soll, müssen häufig gemeinsame Lösungen (auf für Risiken und Abhängigkeiten) im ganzen agilen Team gefunden. Risk-Storming ist ein zum Beispiel bewährtes Verfahren zur gemeinsamen Risikoidentifikation agilen Teams. Planning Poker ist ein weiteres bekanntes Beispiel wie Gamification eingesetzt werden kann, jedoch mit dem Fokus der Bewertung von Aufwänden, Komplexitäten und auch Risiken.

Für das kollaborative Risikomanagement mit Fokus auf Abhängigkeiten in der Softwareentwicklung hat sich Dependency Poker bewährt. Durch die Integration von spielerischen Elementen in die Arbeitsprozesse bleibt das Entwicklungsteam motiviert und engagiert, selbst bei schwierigen und stressigen Aufgaben.

Dependency Poker hilft systematisch und strukturiert (durch die Verwendung von Dependency

Types³) nachvollziehbare Antworten auf die Bewertung von Auswirkungen und Eintrittswahrscheinlichkeiten zu erarbeiten. Durch die Auffächerung der Gegenmaßnahmen in einzelne Lösungsansätze (Dependency Breaker Cards) wird das Finden von Lösungsstrategien transparenter dargestellt und erleichtert somit Entscheidungen im Team.

Zunächst wollen wir erst einen bewährten Ablauf Dependency Poker vorstellen, und diesen dann mit einem Beispiel greifbarer zu machen.

Vorgehen

Für eine optimale Durchführung empfiehlt es sich, Dependency Poker während des Refinement-Prozesses entweder mit dem gesamten agilen Team oder mit ausgewählten Fachspezialisten durchzuführen.

1. Alle Dependency-Type Karten werden auf das ausgewählte Backlog-Element gelegt.
2. Das Team entfernt per Konsens alle Karten bzw. Abhängigkeitstypen, die das Backlog-Item nicht besitzt. Falls es keinen Konsens gibt, darf der Abhängigkeitstyp nicht entfernt werden.
3. Die Abhängigkeitstypen werden besprochen und geclustert. Es ist möglich, dass unterschiedliche Rollen verschiedene Sichtweisen auf die-

selbe Herausforderung haben. Der gleiche Abhängigkeitstyp kann auch mehrfach für ein Item auftreten.

4. Jedes Team-Mitglied wählt die passenden Dependency Breaker Karten für das Risiko-Cluster aus, und das Team deckt alle Karten gleichzeitig auf.
5. Das Team entscheidet sich im letzten Schritt für die erfolgversprechendste Strategie, um die Eintrittswahrscheinlichkeit oder die Auswirkung zu reduzieren. Dies kann zum Beispiel durch ein "Fist-of-Five" Voting stattfinden.
6. Das Risiko und die Lösungsstrategie werden in das Risikoregister oder im Product Backlog eingetragen.

Projektbeispiel

In unserem Projektbeispiel wird das Vorgehen mit dem Gamification-Ansatz angewendet. Dabei handelt es sich um eine komplexe Softwareentwicklung für ein automatisiertes Dispositionssystem am Flughafen, bei der eine zentrale Quelle für tagesaktuelle Flugdaten benötigt wird. Im Refinement, mit dem Ziel, das Minimum Viable Product (MVP) in drei Monaten fertigzustellen, diskutiert das Scrum Team gemeinsam mit dem Projektleiter des Teams und dem Architekten des Flughafens die Anforderungen und potenzielle Abhängigkeiten zu anderen Teams und Produkten⁴.

¹ Zum Thema MVP: https://medium.com/@wps_de/collaborative-modeling-domain-storytelling-as-a-catalyst-for-product-backlogs-d6842dff892b und <https://www.informatik-aktuell.de/entwicklung/methoden/mit-domain-storytelling-ein-minimum-viable-product-entwickeln.html>

² <https://domainstorytelling.org/>

³ Alle notwendigen Karten sind bei unFix kostenlos zu beziehen, nur eine Registrierung ist notwendig und hier zu finden: <https://unfix.com/dependency-breakers> und <https://unfix.com/tips/types-of-dependencies>

⁴ Mehr über das Refinement in der Domäne Flughafen: <https://medium.com/@nilsbert/anforderungsermittlung-in-einer-hochkomplexen-fast-chaotischen-umgebung-das-backlog-refinement-ac5691ca7ce6>

Kollaboratives Risikomanagement mit Dependency Poker: Beschleunigte Analyse von Abhängigkeiten und nachhaltige Lösungsstrategien (Forts.)

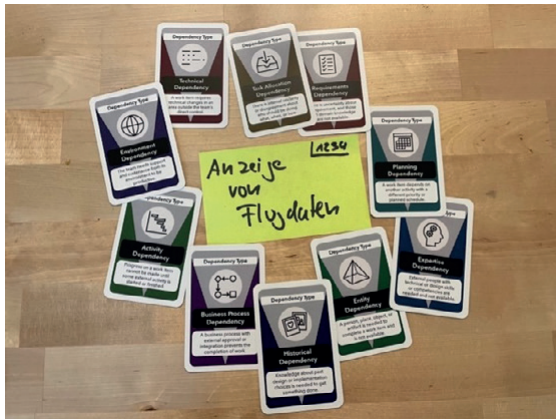


Abbildung 2: Sticky Note des EPIC mit allen Dependency Karten

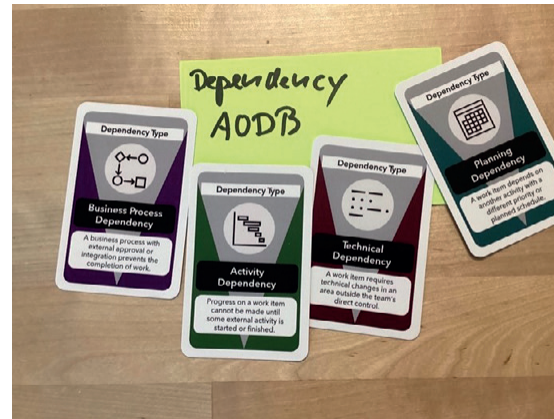


Abbildung 3: Identifiziertes Abhängigkeitscluster



Abbildung 4: Identifiziertes Abhängigkeitscluster mit Lösungsstrategien



Abbildung 5: Risikomatrix

Hierzu wendet das Projektteam das Dependency Poker an, um relevante Abhängigkeiten und Risiken zu identifizieren und zu priorisieren (Abbildung 2). Das Team wählt sorgfältig die relevanten Abhängigkeitstypen aus und diskutiert sie aus verschiedenen Blickwinkeln (Abbildung 3). Dabei identifiziert es die Planning-, Activity-, Business Process- und Technical Dependency, und erlangt ein umfassendes Verständnis der wichtigen Abhängigkeiten und abgeleiteter verbundener Risiken. Während der Diskussion stellt das Team fest, dass alle Typen auf dieselbe Dependency hinweisen, die zu einem Cluster (Abbildung 4) zusammengefasst wird. Schließlich wird deutlich, dass die Daten regelmäßig aus dem zentralen Flugdatensystem (AODB), das von einem anderen Dienstleister entwickelt wird, geladen werden müssen und dies über einen Enterprise Service Bus erfolgen muss.

Die verschiedenen Abhängigkeiten und Risiken werden in eine Risikomatrix (Abbildung 5) eingetragen, um ihre Auswirkungen und Eintrittswahrscheinlichkeiten zu bewerten.

Ein Beispiel für eine solche Abhängigkeit ist, dass komplexe Algorithmen nicht getestet werden können, wenn keine realistischen Testdaten zur Verfügung stehen. Der kritische Pfad des MVPs könnte sich im schlimmsten Fall um zwei Monate verlängern.

Das Projektteam diskutiert nun passende Dependency Breaker für den Cluster „Flugdatenschnittstelle“. Dazu ordnen die Teammitglieder jedem Cluster die entsprechenden Dependency Breaker zu. Jedes Teammitglied bringt seine Sichtweise ein und es werden auch kontroverse Diskussionen geführt.

Die möglichen Dependency Breaker für das Cluster „Flugdatenschnittstelle“ sind:

- Self Service (Das Scrum Team konfiguriert sich über ein Webinterface am ESB die benötigte Schnittstelle selber, indem es die benötigten Daten auswählt und über eine einfache Regel-Engine verknüpft)
- Reteam (Die AODB-Entwickler werden in Sprint #04 direkt im Scrum Team eingebunden)
- Decouple (Ein Schnittstellen-Emulator wird erstellt und genutzt)
- Block (Die Anforderung wird nicht zeitnah in den nächsten Sprints umgesetzt)
- Prioritize (Die Abhängigkeit wird im zentralen Change Board entsprechend hoch eingeordnet)

Kollaboratives Risikomanagement mit Dependency Poker: Beschleunigte Analyse von Abhängigkeiten und nachhaltige Lösungsstrategien (Forts.)

Das gemeinschaftliche Priorisieren und Sortieren der Dependency Breaker führt zu einem Gesamtbild und einem gemeinsamen Verständnis im Projektteam. Auf dieser Basis wird eine Reihenfolge bezüglich der Erfolgswahrscheinlichkeit der Mitigation festgelegt und der wirksamste bzw. vielversprechendste Ansatz wird ausgewählt und im Projekt im Risiko Backlog weiterverfolgt. Durch die Anwendung des Dependency Poker hat jedes Teammitglied die Möglichkeit, aktiv mitzuwirken und sich gehört zu fühlen. Das Projektteam erlangt ein gemeinsames Verständnis für Abhängigkeiten und Risiken, was eine erfolgreiche Zusammenarbeit an Lösungen ermöglicht.

Fazit:

Die Nutzung der Dependency Type und Breaker Karten von unFix⁵ erlaubt eine spielerische Klärung von Abhängigkeiten und Risiken und behält dabei ein sehr strukturiertes Vorgehen im Blick, in dem alle Beteiligten ihre unterschiedlichen Blickwinkel und Sichtweise auf die Problematiken einbringen können. Es entsteht ein transparentes und nachvollziehbares Bild warum, mit welchen und wie mit den Projektrisiken umgegangen werden soll. Durch das systematische Vorgehen und die Nutzung des kollaborativen Schwarmwissens steigt die Erfolgswahrscheinlichkeit in der Behandlung der Risiken signifikant.

Für die Software-Entwicklung optimierter Kartensätze und weitere Variationen für das Vorgehen werden in Kürze auf folgender Webseite zu finden sein: <https://www.dependencypoker.com>

⁵ <https://unfix.com>

Autoren:



Nils Hyoma ,

Diplom-Wirtschaftsinformatiker, im Job Agiler Coach und privat leidenschaftlicher Wasserballspieler. Nils hat Japanologie, internationale BWL und Wirtschaftsinformatik in

Hiroshima und Hamburg studiert.

Novatec Consulting GmbH

nilshyoma@gmail.com

<https://www.linkedin.com/in/nils-hyoma-74a79244/>



Carsten Lill,

Carsten interessiert sich für das Finden und Umsetzen von smarten Lösungen für die Softwareentwicklung in komplexen Umgebungen. Dabei setzt er auf agile Ansätze und umfangreiche Kollaboration

mit allen Stakeholdern für die Anforderungsermittlung. Eine seiner Lieblingsdomänen ist Aviation.

WPS - Workplace Solutions GmbH

carsten.lill@wps.de

<https://de.linkedin.com/in/carsten-lill-b9764622/>

