

Use Case 2.0 – Ivar Jacobson 2011

Lukas Ruckwied*, Manfred Dausmann, Joachim Goll

Fakultät Informationstechnik der Hochschule Esslingen – University of Applied Sciences

Wintersemester 2015/2016

Leichtgewichtig, anpassbar und vielseitig

„Use-Case 2.0, The Guide to Succeeding with Use Cases“ aus dem Jahre 2011 von Ivar Jacobson[1] beschreibt eine Technik, mit der die ehemals schwergewichtigen Use Cases (1.0) an eine agile und iterative Entwicklung angepasst werden können. Innerhalb eines Zyklus einer Iteration ist es kaum möglich, die Abläufe eines klassischen, schwergewichtigen Use Case samt Analyse, Implementierung und Test zu realisieren. Use Case 2.0 beschreibt einen Weg, um die bekannten Use Cases 1.0 in kleine iterative Scheiben zu teilen und damit für eine agile Entwicklung passend zu machen. Der Kern des Konzepts von Use Case 2.0 ist, dass ein Use Case anhand von Stories vertikal in sogenannte Scheiben (engl. slices) aufgeteilt wird, um daraus mehrere kleine Planungseinheiten für die zyklische Auslieferung zu erzeugen. Diese Use Case Slices (dt. Anwendungsscheiben) können in agilen Entwicklungsmethoden den Platz der sogenannten User Stories einnehmen. Die aus Extreme Programming bekannten User Stories sind klein und damit bestens als Planungseinheit für die kurzen Zyklen einer iterativen Entwicklung geeignet. Use Case Slices, die ähnlich wie die User Stories aufgebaut sind und dedizierte Abläufe umfassen, enthalten darüber hinaus eine Referenz auf einen schwergewichtigen Use Case. Durch diese Referenz, wie in Abbildung 1 veranschaulicht, ist es möglich, eine Use Case Slice auf den zugehörigen Use Case zurückzuführen und damit in das klassische Use Case-Diagramm eines Systems einzuordnen. Damit erhält man eine Gesamtschau des Systems und ein gemeinsames Verständnis, auch „Shared Understanding“ genannt, wie das System aussehen soll. Beides ist für eine zielgerichtete Entwicklung eines Systems eine absolute Notwendigkeit. Bei der Verwendung von User Stories hingegen

ist dieses Ziel nicht ohne erheblichen zusätzlichen Aufwand erreichbar.

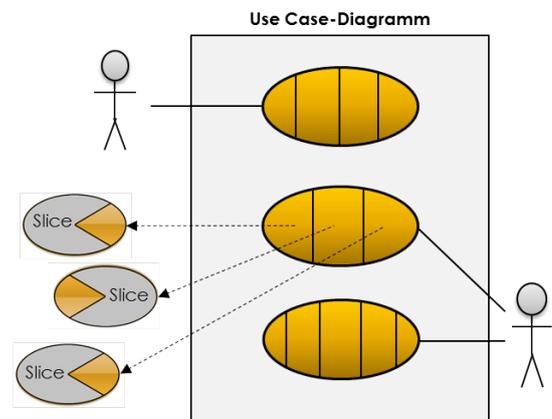


Abbildung 1: Gesamtansicht durch Use Case-Diagramm

Ein Use Case (1.0) sollte entsprechend seiner Abläufe aufgeteilt werden. Diese Aufteilung, auch Schneiden genannt, trennt einen Use Case entlang seiner Abläufe und wird somit „vertikal“ durchgeführt. Es ist zu beachten, dass der resultierende Umfang einer Scheibe in den Rahmen einer Iteration passt. Um möglichst schnell eine erste Use Case Slice für die Implementierung bereitzustellen, reicht es aus, die Story mit dem einfachsten Ablauf (Basisablauf¹) eines Use Case vorzubereiten. In der folgenden Abbildung ist das Schneiden eines Use Case entlang seiner Abläufe visualisiert. Die daraus entstehenden Fragmente werden mit Test Cases und Storys ergänzt. Diese Fragmente (Abläufe) werden mit einer Referenz auf den zugehörigen Use Case versehen und mit dem Zustand ihrer Entwicklung behaftet, um daraus eine planbare und eigenständige Einheit (Slice) für die iterative Entwicklung zu formen.

* Diese Arbeit wurde durchgeführt bei der Firma it-Designers, Esslingen-Zell

¹ Als Basisablauf wird der Ablauf bezeichnet, der den „Gut-Fall“ eines Use Case beschreibt und das gewünschte Ergebnis erbringt.

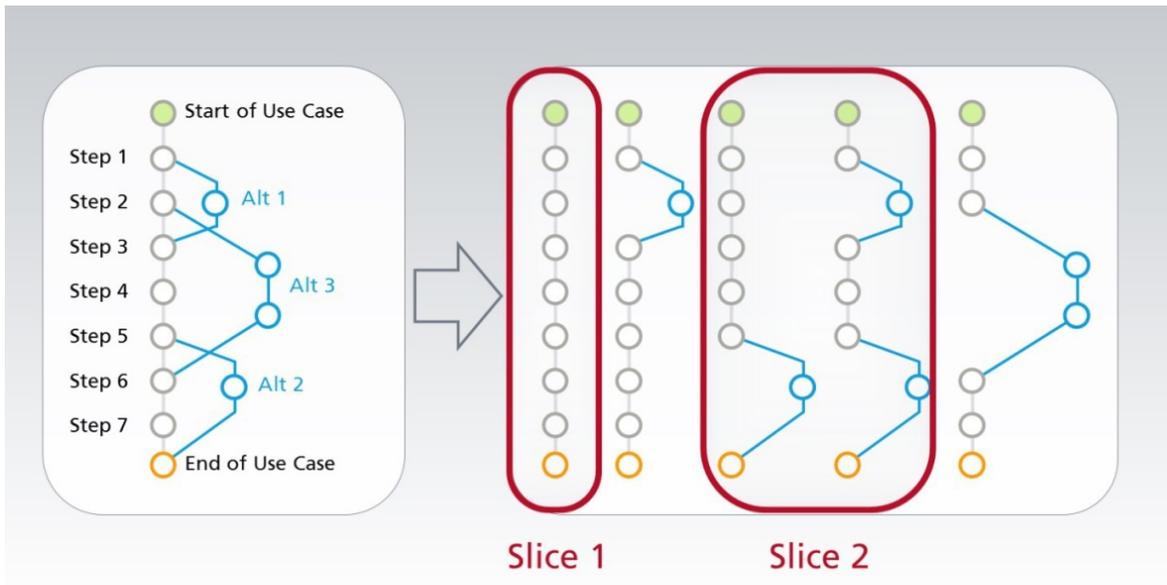


Abbildung 2: Vertikales Schneiden eines Use Case in Slices

Vergleich zu User Story Mapping

Wie schon beschrieben, ähneln sich die Ansätze von Use Case Slices und User Stories. Beide Einheiten sind in dem Zyklus einer Iteration realisierbar. Die Technik Use Case 2.0 möchte bereits von vornherein eine System Vision und eine Gesamtschau des Systems schaffen. Im Falle der Verwendung von User Storys ist User Story Mapping eine weitere Technik, welche ebenfalls ein Gesamtbild erzeugen möchte. Der zusätzliche Aufwand für die Erzeugung eines solchen Gesamtbildes ist allerdings nicht so einfach wie bei Use Case 2.0.

Bewertung

Der Hauptvorteil von Use Case Slices gegenüber User Stories liegt darin, dass die Übersicht und der Fortschritt des Gesamtsystems zu jeder Zeit klar ersichtlich sind. Dieser Ansatz verbindet die Vorteile von spezifikationsorientierten sowie agilen Entwicklungs-

methoden und ist eine sehr gute Alternative zu den bereits aus agilen Praktiken bekannten User Stories. Ob dieser Schritt, spezifikationsorientierte Artefakte mit agilen Prozessen zu verbinden, in der heutigen Softwareentwicklung Interesse weckt und Nutzen findet, bleibt abzuwarten. Use Case 2.0 ist noch relativ jung, muss erst bekannt werden und die bereits auf andere agile Methoden eingespielten Entwickler überzeugen. Die Idee ist gut, jedoch könnte deren Marketing offensiver sein.

[1] I. Jacobson, „Use-Case 2.0 The Definitive Guide,“ Ivar Jacobson International, USA, 2011

Bildquellen:

- Abbildung 1: Eigener Entwurf, keine UML-Notation
- Abbildung 2: <http://www.microtool.de/wie-funktioniert-use-case-2-0/> Zugriff am 15.11.2015